SCHOOL
OF
ENGINEERING
& APPLIED SCIENCE

University of Virginia • Systems Engineering

Tracking Multiple Air Targets in a Sparse Data Environment

A Thesis

Presented to

The Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science in Systems Engineering

Submitted by:

Captain Suzanne M. Oldenburg

May 2000

Approval Sheet

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Systems Engineering.

Suzanne M. Oldenburg

Captain Suzanne M. Oldenburg, Author

This thesis has been read and approved by the Examining Committee:

Dr. Donald E. Brown, Advisor
Department of Systems Engineering

Christina Mastrangelo

Dr. Christina Mastrangelo
Department of Systems Engineering

Dr. Maite Brandt-Pearce
Department of Electrical Engineering

Accepted for the Department of Systems Engineering:

Dr. Donald E. Brown, Chairman
Department of Systems Engineering

Accepted for the School of Engineering and Applied Science:

R. W. Miksad

Dean, School of Engineering and Applied Science
May 2000

Abstract

The main issue with multiple target tracking is associating the observations of a target from one scan with subsequent scans of the target in order to determine which data from one scan are associated with data from previous scans. Once these data points are correlated over several scans, the next step is to determine the trajectory of the underlying targets.

Multiple target tracking (MTT) is essential in military surveillance operations and air tracking control systems. Most MTT systems incorporate linear or piecewise linear algorithms for the filtering and prediction of target positions and finite state Markov Chain techniques. In many instances, data received from one instance to the next consists of a time delay. The greater the time span between data points the more important the ability to be able to estimate the target's position between time spans. Large separations in data points results in a sparse data file requiring the data to be linked together through data fusion in order to capture the complete picture of the target's flight path.

In predicting the target's next location it is necessary for the estimate to be determined prior to receiving the target's next true location. We must be able to process the data in a timely fashion and therefore have an efficient algorithm. This problem is best modeled with time series with the process given in a state-space representation that can handle the multivariate case. The state space model allows the trend and seasonal

component to evolve randomly as a stochastic process rather than deterministically. The state-space model consists of two equations. The observation or measurement equation, Y_t, expresses the n-dimensional observations in vector form. The state equation determines the state at time X_{t+1} in terms of the previous condition and a noise term. The state space model is also referred to as a Markovian or canonical representation of a multivariate time series process. The use of Kalman filtering accommodates a unified method to predict and estimate for all the processes that are given in the state space model.

With advancements in computer technology, other tracking methods are being explored. Bayesian inference is one such method that was not feasible in the past due to the large computational requirements and terrain based tracking uses Bayesian inference. This thesis analyzes two methodologies: the Kalman filtering methodology and the terrain based tracking methodology.

The thesis makes contributions to the knowledge base of multiple target tracking. First, this thesis develops a test environment for multiple target tracking and sparse data. Second, an extension of terrain based tracking is formulated for the application of air targets. Third, the optimization of the Kalman filtering parameters is accomplished with the use of response surface methodology. Finally, testing and evaluation of Kalman filtering and terrain based tracking is examined for multiple targets in both a high data rate environment and the sparse data environment.

Acknowledgments

My first thanks goes to my thesis advisor, Professor Donald E. Brown of the University of Virginia's System's Engineering Department, for his guidance and patience.

I would like to thank Dr. Christina Mastrangelo who has been my mentor since my arrival at the University of Virginia and has always been available to answer my questions. She was instrumental in helping me select the proper courses that I needed to take and was a superb source of information in helping me develop my plan of study, which resulted in allowing me to complete not only my Master's degree but also my PhD course work.

Additionally, I would like to thank Dr. Maite Brandt-Pearce for her time, guidance and resources necessary to conduct my research.

To my parents, Michael and Babette Oldenburg, for their enthusiasm and encouragement during my endeavors.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Symbols

ANOVA	Analysis of Variance
C.V.	Coefficient of Variation
Cor Total	Corrected Total
DF	Degrees of freedom
Dep Mean	Dependent Mean
Error Sigma	Measurement error threshold
KF	Kalman filter
GIS	Geographic Information System
Mean Square	Mean Square Error
MTT	Multiple target tracking
No Go	Rough terrain not traversed by ground vehicles
Open	Open terrain with no road network
PRESS	Predicted Residual Sum of Squares
Pri Road	Primary Road
Prob>F	Probability value > F
Root MSE	Root mean square error
Sec Road	Secondary Road
Sum of Squares	Sum of the squared distance from the mean
TBT	Terrain Based Tracker

1 Introduction

1.1 Motivation

I have an undergraduate degree in Aerospace Engineering and I am an Air Defense Artillery Officer in the United States Army. I have worked with the HAWK and PATRIOT radar systems for the past ten years. This thesis allows me to incorporate my previous areas of study and my current work experience.

1.2 Overview

The goal of tracking is to use observations related to the target to estimate present, future or past states of the target. Target tracking can be accomplished by utilizing both dynamic and static information. Dynamic data are data revealed during the tracking process. Static information consists of knowledge of the terrain, historical patterns of operation, and knowledge of the target's intentions. The basic equations can be modified to account for static information.

Kalman filtering is the traditional method of tracking targets that is a widely accepted tracking algorithm and is found in many of today's radar systems. Kalman filtering uses the technique of gating and state-space representation given prior observations and predicts the next location.

"Bayesian filtering is a recursive form of Baye's theorem. Bayesian trackers are now computationally feasible and extremely flexible in fusing diverse types of information." There are two methods of computation. The batch method is a "Monte Carlo evaluation that replaces stochastic processes by a finite number of sample paths" [Stone, et al., 1999, p. 56]. The second method is a recursive method that takes the previous observation and static information and predicts the future observation. Terrain based tracking uses a Bayesian recursive approach.

The choice of method for filtering and target tracking prediction is the first consideration that must be made when developing a multiple target tracking system. The designer should also consider the data rate in the tracking environment and the states that should be incorporated. These areas will be discussed further.

1.3 Problem Statement

Multiple target tracking (MTT) is essential in military surveillance operations and air tracking control systems. Most MTT systems incorporate linear or piecewise linear algorithms for the filtering and prediction of target positions and finite state Markov chain techniques. In many instances, data received from one instance to the next consists of a time delay. The greater the time span between data points the more important the ability to be able to estimate the target's position between time spans. Large separations

in data points results in a sparse data file requiring the data to be linked together through data fusion in order to capture the complete picture of the target's flight path.

In predicting the target's next location it is necessary for the estimate to be determined prior to receiving the target's next true location. We must be able to process the data in a timely fashion and therefore have an efficient algorithm. This problem is best modeled with time series with the process given in a state-space representation that can handle the multivariate case. The state space model allows the trend and seasonal component to evolve randomly as a stochastic process rather than deterministically. The state-space model consists of two equations. The observation or measurement equation, Y_t, expresses the n-dimensional observations in vector form. The state equation determines the state at time X_{t+1} in terms of the previous condition and a noise term. The state space model is also referred to as a Markovian or canonical representation of a multivariate time series process.

The use of Kalman filtering accommodates a unified method to predict and estimate for all the processes that are given in the state space model. When the time between scans becomes large, 10, 20 and 40 second time intervals for the air target data, the result is a sparse data set resulting in a possibly higher error rate in predicting the target's subsequent locations. With the use of Kalman filtering and terrain based tracking, targets will be examined and tracked given sparse data.

The main issue with sparse target tracking is associating the observations of a target from one scan with subsequent scans of the target in order to determine which data from one scan are associated with data from previous scans. A sparse data environment is defined as one that has a low data rate. These data points are correlated over several scans in order to determine the trajectory of the underlying targets. Because of the low data rate, the traditional methods of Kalman filtering are not as successful due to the non-Gaussian nature of this data environment.

1.4 Approach

The proposed research will provide an estimation of multiple target tracking given air track data files from the National Training Center utilizing the methodologies of Kalman filtering and terrain based tracking in order to determine the most efficient methodology in multiple target tracking prediction. The same data files will then be made sparse by using data points every 10, 20, and 40 seconds. These methods will be applied to the modified data file and the results examined for the sparse data scenario to find the most efficient methodology.

2 Background

2.1 Historical

The first radar systems were introduced to the military as they faced the outbreak of World War II. Having very little experience with the technology, scientists were asked to assist in the use of the newly developed radar to locate enemy aircraft. It was soon found that radar was temperamental and did not function as they had in laboratory conditions. In September 1940, the British physicist, P.M.S. Blackett, brought together a team of scientists to study the systems in the field. "This type of scientific activity became known as 'Operational Research' because the first studies were devoted to the operational use of radar" [Ackoff, 1963, p.7].

Radar has proven to be an invaluable technology both for military and civilian use and remains a major area of investigation in order to improve their capabilities. Most recently, the problem of tracking targets at longer ranges introduces the sparse data tracking problem.

Target tracking and prediction are needed in a radar system due to the scan nature of the radar beam. At time T_1 the radar beam detects multiple targets. Subsequent scans continue to detect multiple targets. It is necessary to correlate these tracks in order to

determine if the tracks detected initially are associated with the same tracks detected during subsequent scans. The correlation problem is more difficult for long-range sensors that have longer scan times. These sensors can also fail to detect targets between scans. This problem is known as sparse data tracking and is vital for new sensor systems employed in both military and civilian contexts such as target interception and surveillance operations and air traffic control. Military radar needs to be able to determine the number of targets and be able to differentiate the targets for interception. If target locations can be predicted accurately ahead of time, the possibility of incorrectly identifying targets can be greatly reduced. For air traffic control it is necessary in order to prevent collisions and to be able to associate the targets' velocities.

Most of today's military and civilian radar use the Kalman filtering algorithm for tracking targets. Some of these radars are summarized in Table 2.1.

Table 2.1 Radars using KF Algorithm

Radar	Purpose
PATRIOT	Wide Angle electronically scanned phased array – Land Based
AEGIS	Wide Angle electronically scanned phased array - Shipboard
HAWK GPS-22 HiPAR	Fan beam radar
AN/SPS-49	Fan beam shipboard track-while-scan
AN/TPQ-37	Long ranged electronically scanned phased array
THAAD	X-band active array system
ASR-11	Digital Airport Surveillance DOD and FAA Track-while-scan
ASR-23SS	L-Band Airport Surveillance Track-while-scan

The radar systems described in Table 2.1 work best when in a high signal-to-noise ratio environment. A high signal-to-noise ratio is a high data rate environment with minimum noise. The signal-to-noise ratio is a random variable and is best modeled as a stochastic process [Washburn, 1982]. There is no universal best choice for a model when developing a radar algorithm. Therefore there are a variety of radar systems built for specific uses as shown in Table 2.1. Part of the consideration of course in developing a new radar system is "cost, time, data availability and communicability of results" [Washburn, 1982, p. 3-8].

2.2 Objectives

This thesis will examine multiple target data sets and apply the tracking algorithms of Kalman filtering and terrain based tracking in order to determine the best approach for an efficient yet accurate algorithm. The sparse data environment will also be explored and results will be reported on both the use of Kalman filtering and terrain based tracking.

2.3 Related Research

2.3.1 United States Navy, Nodestar System

The Nodestar system was developed as a multiple target correlator tracker for the Navy's underwater surveillance system as part of the Spotlight Advanced Technology Demonstration (ADT) [Stone, et al., 1999]. At the ADT, Nodestar functioned as a theater wide, multiple target, and correlator tracker for submarines. Nodestar is capable of processing different forms of information on multiple targets at sufficient speeds for surface ship and submarine applications.

Nodestar's discrete state space has six or more dimensions. Position is represented in latitude, longitude and depth and operates in spherical coordinates. The velocities are represented in two dimensions, and are considered satisfactory for subs and surface ship problems. The sixth dimension is an attribute consisting of the acoustic and radar characteristic. Nodestar's motion updating is accomplished with a discrete space Markov chain motion model. The Nodestar system is a contact based, discrete, non-linear, non-Gaussian, multiple target, Bayesian tracker.

Underwater surveillance systems are typically capable of handling low signal-to-noise ratio environments. As the depth changes, so does the water temperature therefore causing changes in the acoustic signals. This tracker uses the surrounding terrain, water

and the presence of landmasses. The detection of landmasses acts as negative information in this application. It is noted that when tracking submarines, the depth beneath the waters surface is approximately 100 nautical miles. At this depth, the effects of the earth's curvature are negligible and the assumption of an approximately flat earth model is appropriate [Stone, et. al., 1999].

2.3.2 United States Coast Guard, Computer Assisted Search and Rescue Planner (CASP)

CASP 2.0 was developed by Wagner Associates for the U.S. Coast Guard and is a replacement search and rescue planning system. CASP 2.0 models targets before and after distress through paths created by a Monte Carlo model [Wagner, 1999].

CASP is a non-linear Bayesian tracker. CASP produces a probability distribution for target locations as a function of time to project one day ahead in order to plan for the next day's search. CASP uses concepts and algorithms from search theory and incorporates negative information, i.e. keeps track of the areas where the search found no targets. This negative information is used to compute posterior distributions for target locations. This stochastic process predicts the targets position and motion over time [Stone, et al., 1999].

In addition, Wagner Associates demonstrated how track-to-track algorithms efficiently combine simulated outputs from a combination of sensors to produce a correlated picture for use in the United States Air Force for AWACS multi-sensor integration.

2.3.3 Terrain Based Tracker

Work conducted by P.O. Nougues and Professor D.E. Brown developed a terrain based tracker to better predict the movement of ground targets over terrain. Terrain based tracking relates target tracking in terrain by "determining the best estimate of the probability density function of target location for successive time increments by combining the greatest amount of information available on terrain characteristics and target behaviour" [Nougues et al., 1995]. This methodology will be applied to air targets to determine if terrain based tracking can be applied to aircraft that take advantage of the terrain to determine their flight path. For instance, a target flying nap of the earth takes advantage of geographic features such as a mountain ranges, roads, rivers, etc. In addition, the relationship between targets flying over terrain features such as mountainous areas versus open terrain fields will be examined. Terrain based tracking will be examined to determine targets that utilize the terrain and will be used to predict the target's flight path.

3 Kalman Filtering (KF) Algorithm

The traditional method for tracking targets utilizes the Kalman filtering algorithm. Kalman filtering grew in popularity due to its capability of being applied to both stationary and non-stationary problems and has a wide variety of applications. Kalman filtering has become especially important in the Aerospace Engineering field. "The Kalman filter has become the standard tool for error analysis, design of data processing algorithms, navigation systems, Global Positioning System and satellite attitude control and attitude determination, as well as interpolation, extrapolation and smoothing problems" [Space Analytics Associates, 1999].

The Kalman filter is also applied to multiple target tracking, error estimation in gyro systems and tactical ballistic missile positioning. The Kalman filter is very effective in helping obtain estimates of impact points and launcher locations in the case of SCUD, intermediate range ballistic missiles, IRBM, and intercontinental ballistic missiles, ICBM. "A future use of Kalman filters is in the 'smart' highway systems where it will be used to conduct the movement of cars traveling at high speeds in tightly bunched groups" [Cipra, 1993].

The research proposed investigates the Kalman filtering method and compares the results to terrain based tracking and also determines how well Kalman filtering accommodates a sparse data environment.

3.1 Problem Statement

Target tracking and prediction are needed in radar systems due to the scan nature of the radar beam. At time T_1 the radar beam detects two targets. The next scan, occurring at time $T_1 + t$, detects two targets. Are these targets the same two targets detected on the first scan? In civilian air traffic control, it is necessary for the air traffic controller to know the number of targets present in order to prevent collisions and to be able to associate the targets' velocities. Military radar needs to know the number of targets and be able differentiate the targets for interception. If target locations can be predicted accurately ahead of time, the possibility of incorrectly identifying targets can be greatly reduced.

3.2 State Space Model

In order to model this problem in time series, the process can be given in state-space representation. The state-space model consists of two equations. Equation 3.1 shows the observation or measurement equation, Y_t, which expresses the n-dimensional observations in vector form.

Equation 3.1 Measurement Equation in Matrix Notation

$$Y_t = MX_t + N_t$$

Where:

$M = [I_r\, 0]$ Observation Matrix

r = first r variables are the current values

N = Observation error

Equation 3.2 shows the observed values X_t in the state vector Z_t:

Equation 3.2 Observation Matrix

$$X_t = [I_r\ 0]\ Z_t$$

The state equation determines the state at time X_{t+1} in terms of the previous condition and a noise term. Modeling state space makes use of a state vector and a state transition equation. The state vector is represented by Z_t, Equation 3.3, of dimension s, where the first r variables consist of the current variables, X_t. The s-r variables consist of the predicted variables, $X_{t+k/t}$.

Equation 3.3 State Transition Equation in Matrix Notation

$$Z_{t+1} = F Z_t + Ge_{t+1}$$

Where:

F = s x s transition matrix which determines the dynamic properties of the model.

G = s x r input matrix determines the variance structure of the transition equation.

e_t = sequence of independent, identically, normally distributed random vectors with dimension r with zero mean and covariance matrix Σ_{ee}. The variable e_t also refers to the random error.

For the air track data we are only looking at the two dimensional problem.

Figure 3.1 shows the components of the state vector. Zt consists of the position vectors X and Y, velocity vectors Vx and Vy, and the predicted position and velocity vectors:

$$Zt := \begin{bmatrix} X \\ Y \\ Vx \\ Vy \\ Xp \\ Yp \\ Vxp \\ Vyp \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}X\\Y\\Vx\\Vy\end{matrix}}\right\} \boxed{\text{Measured data}} \\ \left.\vphantom{\begin{matrix}Xp\\Yp\\Vxp\\Vyp\end{matrix}}\right\} \boxed{\text{Predicted data}} \end{matrix}$$

Figure 3.1 The State Vector

The state vector summarizes the data from past to present values within the time series relevant to estimating the future values of the series. The observed time series is then represented as a linear combination of the state variables.

Once the state vector is defined, the state space model is fit to the data. F, G and Σ are estimated by approximating the maximum likelihood. Restrictions are placed on the F matrix for the air track data set to impose the restrictions on the distance vectors associated with the time lapse or lag in the radar observation of the target. Figure 3.2 summarizes the restrictions on the F matrix.

$$\left[\begin{array}{cccccccccccc}
1 & 0 & 0 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & T & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
a1 & b1 & c1 & d1 & e1 & f1 & g1 & h1 & i1 & j1 & k1 & l1 \\
a2 & b2 & c2 & d2 & e2 & f2 & g2 & h2 & i2 & j2 & k2 & l2 \\
a3 & b3 & c3 & d3 & e3 & f3 & g3 & h3 & i3 & j3 & k3 & l3 \\
a4 & b4 & c4 & d4 & e4 & f4 & g4 & h4 & i4 & j4 & k4 & l4 \\
a5 & b5 & c5 & d5 & e5 & f5 & g5 & h5 & i5 & j5 & k5 & l5 \\
a6 & b6 & c6 & d6 & e6 & f6 & g6 & h6 & i6 & j6 & k6 & l6
\end{array}\right]
\begin{array}{l}
\left.\vphantom{\begin{array}{c}1\\1\\1\\1\\1\\1\end{array}}\right\}\ \boxed{\text{Restrictions on F}}\\
\left.\vphantom{\begin{array}{c}1\\1\\1\\1\\1\\1\end{array}}\right\}\ \boxed{\text{Parameters to be estimated}}
\end{array}$$

Figure 3.2 The Transition Matrix

After the parameters are estimated, the forecasts are made from the state-space model using Kalman filtering. The Kalman filtering method accommodates a unified method to predict and estimate for all the processes that are given in the state-space model.

3.3 Types of Kalman Filters

The Kalman filter deals with the general problem of first order, discrete time controlled processes. Variations of this are described below:

1. Extended Kalman Filter (EKF) – applied to non-linear systems by linearizing about the current mean and covariance.

2. Fixed Lag Kalman Smoother (FLKS) – an approach to perform retrospective linear data assimilation.

3. Extended Fixed Lag Kalman Smoother – extension of FLKS applied to non-linear dynamics and observation processes.

Other types of filters and some special filters include:

- Weiner filter
- Fading-memory polynomial filter
- Expanding–memory polynomial filter
- Kalman filter
- Baye's filter
- Least-squares filter
- Benedict-Bordner filter
- Lumped filter
- Discounted least–squares g-h filter
- Critically damped g-h filter
- Growing-memory filter

These filters all differ by how the weighting factors are selected. Some are dependent on the number of observations as it is expected that as the number of observations increases, the prediction becomes more accurate. The weighting factors therefore initially are small. These filters can also be fixed coefficient filters and are used primarily when the sampling interval is small or there is a multiple target environment.

3.4 Benefits of Kalman Filter:

The benefits of Kalman filtering are summarized as found in Brookner, 1998:

1. Provides running measure of accuracy of predicted position needed for weapon kill probability calculations. (Impact point prediction)
2. Permits optimum handling of measurements of accuracy that varies with n, missed measurements, and non-equal time measurements
3. Allows optimum use of a priori information if available
4. Permits target dynamics to be used directly to optimize filter parameters
5. Addition of random-velocity variable, which forces the Kalman filter to always be stable

3.5 Kalman Filtering Assumptions

The following assumptions for Kalman filtering are the assumptions applied to this problem.

1. The initial state X_1 is uncorrelated with all noise terms.
2. The noise is all Gaussian white noise with zero mean
3. The targets velocity is constant
4. The radars scan-to-scan period = T. For the full data set, T =1 second.
5. Atmospheric drag (β) on the target is negligible

In summary, the two critical assumptions of Kalman filtering are a linear model with Gaussian measurement error and a discrete model with a 2σ uncertainty ellipse.

Distinction is made with respect to the Kalman filtering motion assumption. This assumption states that "… Markov transition functions applied to Gaussian distributions on the target state space, [they] produce another Gaussian distribution" [Stone, et al., 1999, p.66].

Finally, assumptions for Kalman filtering measurements are as follows [Stone et al., 1999, p.68]:

1 All observations must be linear functions of the target state

2. The observation errors are Gaussian

3. Y_t must satisfy the measurement Equation 3.1

4. Measurement error is independent of X_t

These assumptions guarantee that if the distribution of X_t is Gaussian prior to the observation, it is Gaussian posterior to the observation [Stone, et al. 1999].

Kalman filtering can consider the following three models. Our model is the constant velocity model.

1. Constant velocity model
2. Constant acceleration model
3. Variance of velocity distribution is bounded

The Kalman filter prediction Equation 3.4 and Equation 3.5, for a target being tracked by only target velocity and position in one dimension, are shown below:

Equation 3.4 System Dynamic Position Model

$$x_{n+1}=x_n+T\dot{x}_n$$

Equation 3.5 System Dynamic Velocity Model

$$\dot{x}_{n+1}=\dot{x}_n+u_n$$

Where:

u_n = random velocity jump just prior to time n+1.

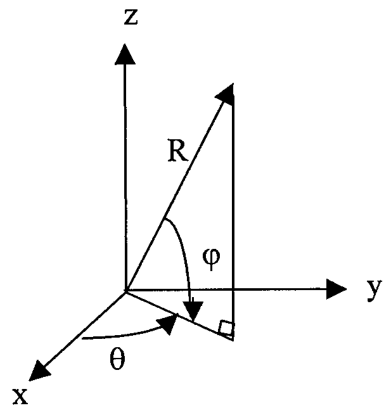
x_n = targets range at scan n

$\dot{x}_n$= target velocity

T = scan-to-scan period.

3.5.1 Radar Observations

The radar makes the target measurements in polar coordinates. These measurements are transformed to rectangular coordinates in order for the tracking filter to operate and then back into polar coordinates after each prediction is made so that the next observation can be made in the measurement space. The data set from the National Training Center (NTC) was provided in rectangular coordinates. This data set has therefore already been transformed from the measurement polar coordinates. This relationship is shown in Figure 3.3.



$$R = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = ARCTAN\left(\frac{y}{x}\right)$$

$$\phi = ARCSIN\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$$

Figure 3.3 Polar Coordinate Relationship

There are advantages and disadvantages to using Cartesian coordinate systems. Two disadvantages are the use of the range measurement and the coupling of the measurement errors. If a target is flying in an electronic countermeasure environment, an environment that can jam the radar, the range data will not be able to be obtained thus causing the radar to receive no target information and hence result in sparse data.

For the purpose of air target tracking, our targets fly lower than 100 miles above the earth's surface. As discussed with the Nodestar system, a flat earth approximation is also assumed to be appropriate.

In describing the target motion, a constant velocity model is assumed. In the real world this is more than likely an invalid assumption, however this assumption is made for this research. The uncertainty in the target's trajectory such as the target accelerating or

turning at any time, is a feature of the Kalman filter that allows us to accurately predict the target's motion taking this into consideration.

3.5.2 Gating and Correlation Threshold

Gating is used to eliminate observations that are not feasible for data association given a target within the gate. If there is a single observation within the gate and this gate does not overlap with another target's gate, then a correlation is made between the target in the gate and the next observation found within the gate. If there are multiple observations within a single gate, further analysis of the observations is made before a correlation is formed. Gating logic is therefore used to minimize the number of pairings that must be considered for data association.

In performing track correlation, gating acts as a course test in order to classify the target into one of two categories. The first category is that which makes the target a potential candidate for updating if that track satisfies the criteria for one or more tracks. The second category is for tracks that do not satisfy the gates for existing tracks and therefore a new track is initiated [Blackmann, 1986].

In using the Kalman filtering algorithm a gate size was selected such that the best results were obtained for the given data set. The gate size was minimized along with the correlation threshold parameter in order to maintain the greatest track correlation and the

least number of dead tracks generated. Also, it was important to maintain a reasonable gate size so that performance does not change with an increase in the number of tracks per data set.

3.5.3 Parameter Estimation

The parameter file used for Kalman filtering is shown in detail in Appendix A. These parameters are as follows:

1. Q = Assumed known covariance matrix Equation 3.6.

"By appropriately selecting Q, we will minimize the effects of miscorrelation and also reduce the unstable tracking region" [Blackman, p. 106]

Equation 3.6 Covariance Matrix

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_{\delta v}^2 \end{bmatrix}$$

2. KG = Rectangular Kalman gating constant which sets the size of the gate
3. Error Sigma = Standard deviation of filtered estimation error. This parameter is a threshold of how much deviation outside the gate we allow and still consider a track to correlate with the previous observation.

The Kalman filter parameters of Q, covariance, and KG, gating, were optimized along with the error sigma using response surface methodology and Design Expert software. A response surface utilizing three factors set at three levels, a 3^3 design, was used to optimize these parameters. Three responses were used: percent correct correlation, mean prediction percentile and the number of dead tracks. This process is discussed further.

4 Response Surface Methodology and KF Parameter Optimization

In order to compare Kalman filtering to terrain based tracking, it is important to determine the best set of parameters to utilize in Kalman filtering so that the best results for Kalman filtering can be achieved. In this section we explore the Kalman filtering parameters and find the best operating levels of these parameters to optimize the response variables.

4.1 Design of Experiment

A user defined model using Design Expert software was created. The model consisted of three factors using three levels for each factor and there were three responses recorded for each run. Table 4.1 shows the design of the experiment in the actual factors.

Table 4.1 Design of Experiment Actual Factors

Run	Q	KG	Error Sigma
1	0.1	1.5	20
2	0.1	1.5	40
3	0.1	1.5	80
4	0.1	2.5	20
5	0.1	2.5	40
6	0.1	2.5	80
7	0.1	5	20
8	0.1	5	40
9	0.1	5	80
10	1.5	1.5	20
11	1.5	1.5	40
12	1.5	1.5	80
13	1.5	2.5	20
14	1.5	2.5	40
15	1.5	2.5	80
16	1.5	5	20
17	1.5	5	40
18	1.5	5	80
19	2	1.5	20
20	2	1.5	40
21	2	1.5	80
22	2	2.5	20
23	2	2.5	40
24	2	2.5	80
25	2	5	20
26	2	5	40
27	2	5	80

The design is summarized as a response surface, user defined, 3^3 design, with 27 experiments. The design model is quadratic. The actual factors were scaled and centered. Table 4.2 shows the design of the experiment in terms of the coded factors.

Table 4.2 Design of Experiment Coded factors

Run	Q	KG	Error Sigma
1	-1	-1	-1
2	-1	-1	-0.33333333
3	-1	-1	1
4	-1	-0.428571	-1
5	-1	-0.428571	-0.33333333
6	-1	-0.428571	1
7	-1	1	-1
8	-1	1	-0.33333333
9	-1	1	1
10	0.473684	-1	-1
11	0.473684	-1	-0.33333333
12	0.473684	-1	1
13	0.473684	-0.428571	-1
14	0.473684	-0.428571	-0.33333333
15	0.473684	-0.428571	1
16	0.473684	1	-1
17	0.473684	1	-0.33333333
18	0.473684	1	1
19	1	-1	-1
20	1	-1	-0.33333333
21	1	-1	1
22	1	-0.428571	-1
23	1	-0.428571	-0.33333333
24	1	-0.428571	1
25	1	1	-1
26	1	1	-0.33333333
27	1	1	1

Table 4.3 summarizes the response variables and the three factors used in the model.

Table 4.3 Design Summary

Response	Name	Obs	Minimum	Maximum	Model
Y1	% Corr	27	89.31	97.00	Reduced Quadratic
Y2	Mean pred	27	38.06	89.43	Reduced Quadratic
Y3	#Dead Tracks	27	214.00	1222.00	Reduced Quadratic

Factor	Name	Low Actual	High Actual	Low Coded	High Coded
A	Q	0.100	2.00	-1.000	1.000
B	KG	1.50	5.00	-1.000	1.000
C	Error	20.00	80.00	-1.000	1.000

4.2 Model Selection

Three models were evaluated, one for each response variable. Stepwise regression was performed on the full model with alpha to enter equals 0.100 and alpha to exit equals 0.100. An analysis was conducted on each of the factors to determine if they were significant in the model. Once the appropriate model was selected, an optimization on the response variable was performed to find the optimal operating conditions for Kalman filtering.

A summary of the three factors and their respective actual high and low levels are summarized in Table 4.4.

Table 4.4 Summary of Factor Levels

Factor	-1 Level	+1 Level
Q	.10	2
KG	1.5	5
Error Sigma	20	80

4.2.1 Response = Mean Prediction Percentile

The results from stepwise regression and the model selected for the mean prediction percentile is as follows:

	Coefficient Estimate	Prob > |t|	R-Squared	MSE
Error Sigma	-12.40	<0.0001	0.6072	74.20
KG	-7.22	<0.0001	0.8182	35.78
Q	4.73	<0.0001	0.9099	18.51
KG * Error Sigma	-4.48	<0.0001	0.9660	7.30
Q^2	-3.37	0.0184	0.9741	5.84

The model is a good fit for the data as can be seen by evaluating Table 4.5. The adjusted R^2 for this model is 0.968.

Table 4.5 ANOVA for Mean Prediction Percentile

Source	Sum of Squares	DF	Mean Square	F Value	Prob > F
Model	4600.47	5	920.09	157.68	< 0.0001
Residual	122.54	21	5.84		
Cor Total	4723.01	26			

Root MSE	2.42	R-Squared	0.9741	
Dep Mean	73.63	Adj R-Squared	0.9679	
C.V.	3.28	Pred R-Squared	0.9559	
PRESS	208.48	Adeq Precision	44.374	Desire > 4

All three factors are significant as determined by the t-test. Equation 4.1 shows the final equation for this model in terms of the actual factors. The parameter Q contributes most to the model of mean prediction percentile.

Equation 4.1 Mean Prediction Percentile in Terms of Actual Factors

Mean predection percentile $= 86.24 + 12.42\ Q - 0.15\ KG - 0.16\ \text{Error Sigma} - 3.73\ Q^2 - 0.085\ KG * \text{Error Sigma}$

Next, the model was examined to determine at what level for the three factors is the maximum mean prediction percentile obtained. Figure 4.1 shows the response surface and the relationship of Q, KG and mean prediction percentile.

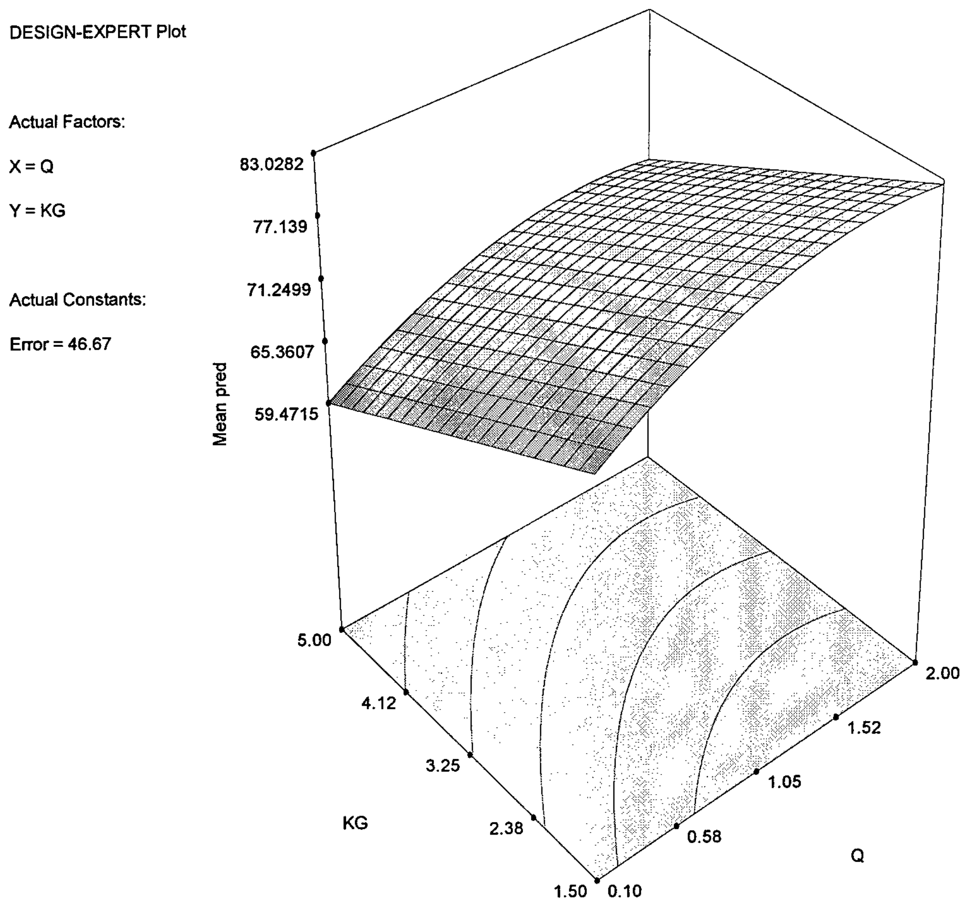


Figure 4.1 Mean Prediction Percentile Response Surface

An optimization of all three parameters gives us the solutions in Table 4.6.

Table 4.6 Mean Prediction Percentile Optimal Solutions

	Q	KG	Error Sigma	Mean pred	Desirability
1	1.64	1.57	20.48	90.3703	1.000
2	2.00	1.76	20.33	89.6422	1.000
3	1.67	1.57	20.73	90.298	1.000
4	1.64	1.61	23.13	89.5208	1.000
5	1.33	1.90	20.08	89.4638	1.000
6	1.80	1.96	20.64	89.5108	1.000
7	1.52	1.51	23.82	89.4411	1.000
8	1.90	1.70	21.53	89.6027	1.000
9	1.63	1.61	20.48	90.2883	1.000
10	0.70	1.50	46.03	79.7832	0.812

The desireability function, d, for a response being maximized is such that d = 1, or as close to 1 as possible, for any predicted value greater than a target response value.

4.2.2 Response = Percent Correct Correlation

The results from stepwise regression and the model selected for the percent correct correlation is as follows:

Added	Coefficient Estimate	Prob > |t|	R-Squared	MSE
KG	2.07	<0.0001	0.5628	2.55
Error Sigma	1.28	<0.0001	0.7739	1.37
KG * Error Sigma	-0.85	0.0057	0.8391	1.02
Error Sigma^2	-1.02	0.0268	0.8719	0.85

The significant result when modeling percent correct correlation is that the term Q was found to be insignificant in the model to predict the percent correct correlation (This factor was found to contribute most to the mean prediction percentile shown in section 4.2.1.). The model was found to be a good fit. Table 4.7 shows an adjusted R^2= 0.849.

Table 4.7 ANOVA for Percent Correct Correlation

Source	Sum of Squares	DF	Mean Square	F Value	Prob > F
Model	127.24	4	31.81	37.43	< 0.0001
Residual	18.70	22	0.85		
Cor Total	145.93	26			

Root MSE	0.92	R-Squared	0.8719	
Dep Mean	93.97	Adj R-Squared	0.8486	
C.V.	0.98	Pred R-Squared	0.8216	
PRESS	26.04	Adeq Precision	17.242	Desire > 4

It was found that only two factors, KG and error sigma, are significant as determined by the t-test. This model in terms of the actual factors is shown in Equation 4.2. The KG parameter contributes most to this model for the response percent correctly correlated.

Equation 4.2 Percent Correctly Correlated in Terms of Actual Factors

$$\% \text{ Correct Corr} = 83.88 + 1.94\ \text{KG} + 0.21\ \text{Error Sigma} - 0.001137\ \text{Error Sigma}^2 - 0.016\ \text{KG} * \text{Error Sigma}$$

Figure 4.2 shows that as KG varies, the maximum percent correct correlation does not change significantly as Q varies.

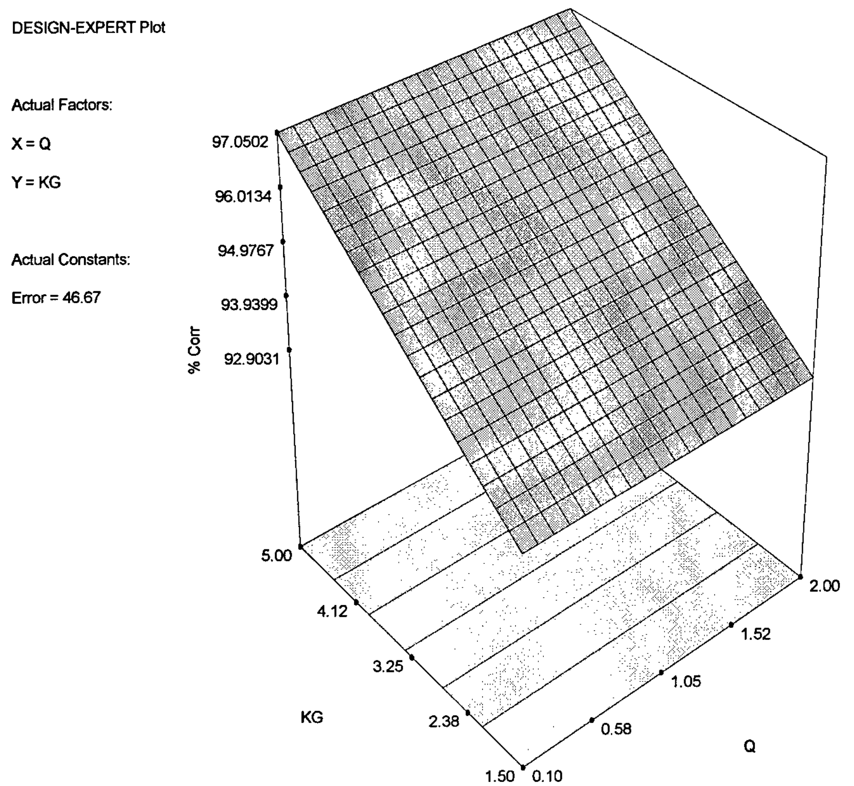


Figure 4.2 Percent Correctly Correlated Response Surface

The optimal candidate solutions for percent correct correlation are summarized in Table 4.8.

Table 4.8 Percent Correct CorrelationOptimal Solutions

	Q*	KG	Error Sigma	% Corr	Desirability
1	*0.46*	4.92	59.63	97.0545	1.000
2	*1.02*	4.96	46.97	97.0127	1.000
3	*2.00*	4.89	61.57	97.014	1.000
4	*0.84*	4.90	59.22	97.0377	1.000
5	*0.15*	4.92	63.54	97.0117	1.000
6	*0.89*	4.98	64.93	97.0438	1.000
7	*0.82*	4.94	51.66	97.0615	1.000
8	*0.18*	5.00	60.40	97.1268	1.000
9	*1.80*	4.95	48.36	97.0278	1.000
10	*1.19*	4.86	56.29	97.0077	1.000

*Has no effect on optimization results.

4.2.3 Response = Number of Dead Tracks

The parameter Q is added to this model because through the stepwise regression procedure it was found that Q^2 was significant in the model. It is desired that a hierarchical model is maintained, therefore Q must remain in the model. The model selcted for the percent correct correlation is as follows:

Added	Coefficient Estimate	Prob > |t|	R-Squared	MSE
KG	-258.79	<0.0001	0.5356	44373.35
Error Sigma	-200.04	<0.0001	0.8483	15100.02
KG * Error Sigma	81.41	0.0126	0.8849	11951.18
Error Sigma^2	127.96	0.0089	0.9163	9092.16
Q^2	-98.42	0.0471	0.9309	7860.33
Q	-35.78	0.0868	0.9405	7101.30

Table 4.9 verifies that the model selected is appropriate given the adjusted $R^2 = 0.923$.

Table 4.9 ANOVA for Number of Dead Tracks

Source	Sum of Squares	DF	Mean Square	F Value	Prob > F
Model	2.247E+06	6	3.744E+05	52.73	< 0.0001
Residual	1.420E+05	20	7101.30		
Cor Total	2.389E+06	26			
Root MSE	84.27		R-Squared	0.9405	
Dep Mean	632.44		Adj R-Squared	0.9227	
C.V.	13.32		Pred R-Squared	0.8977	
PRESS	2.443E+05		Adeq Precision	23.604	Desire > 4

All factors are significant as determined by the t-test. The final equation, Equation 4.3, for this model in terms of the actual factors is as follows:

Equation 4.3 Number of Dead Tracks in Terms of Actual Factors

#Dead Tracks = 1876.14 + 242.18 Q –220.24 KG –25.94 Error Sigma – 133.26Q^2 + 0.14 Error Sigma^2 + 1.55 KG * Error Sigma

Figure 4.3 shows the relationship between KG, Q and the number of dead tracks.

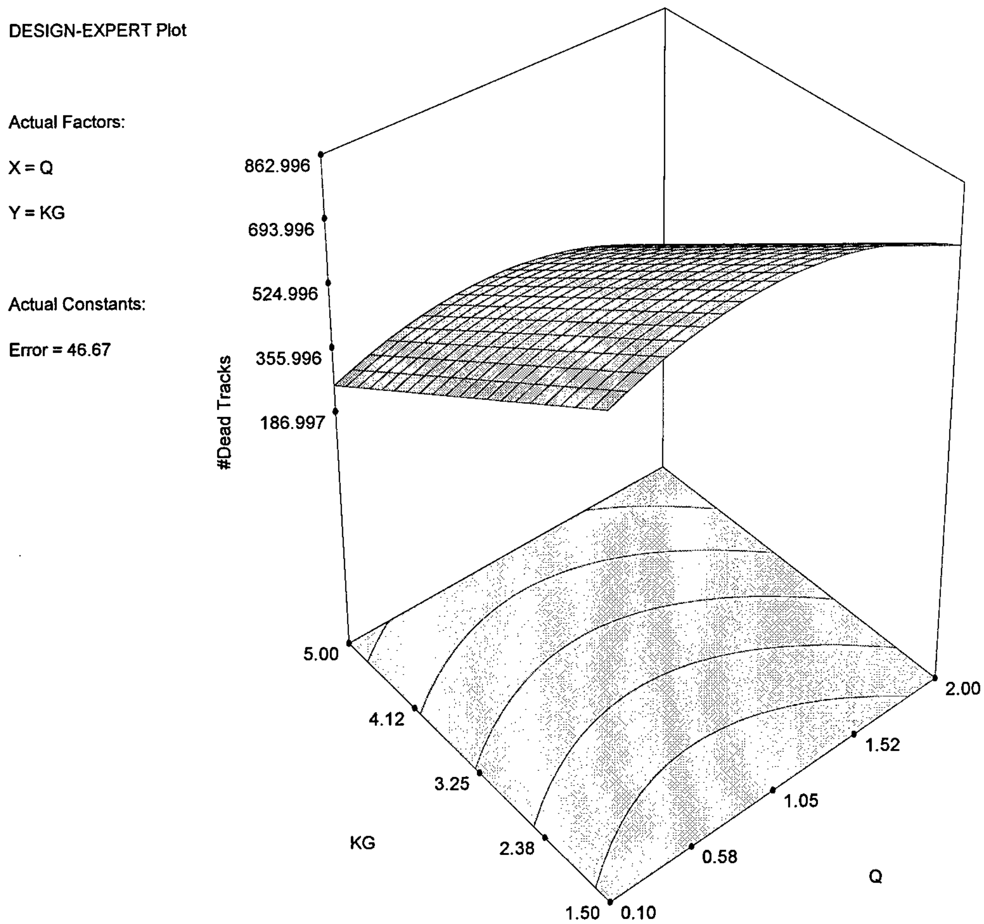


Figure 4.3 Number of Dead Tracks Response Surface

The criteria for parameter optimization is:

1. Q, KG, and error sigma are within range
2. Number of dead tracks is minimized

The desireability function, d, for a response being minimized is such that d = 1 for any predicted value less than a target response value. This criteria results in the solutions found in Table 4.10.

Table 4.10 Number of Dead Tracks Optimal Solutions

	Q	KG	Error Sigma	#Dead Tracks	Desirability
1	2.00	5.00	64.58	144.596	1.000
2	2.00	5.00	62.52	144.97	1.000
3	2.00	5.00	62.78	144.975	1.000
4	1.99	5.00	64.92	146.221	0.999
5	2.00	4.97	64.91	148.524	0.997
6	0.10	5.00	63.99	216.062	0.934
7	0.10	4.96	61.23	221.564	0.929
8	1.63	5.00	79.84	269.538	0.884
9	0.45	5.00	62.92	275.597	0.879

4.3 Final Parameter Selection

In optimizing all three responses, the optimal parameters for Kalman filtering given the high data rate are shown in Table 4.11.

Table 4.11 KF Optimal Solutions

	Q	KG	Error Sigma	% Corr	Mean pred	#Dead Tracks	Desirability
1	1.99	4.65	55.15	96.7869	64.9258	205.998	1.000
2	1.97	4.93	46.11	96.9584	68.8418	209.168	1.000
3	1.98	4.56	62.68	96.6956	61.2593	203.404	1.000
4	1.98	4.98	44.70	96.9773	69.4571	207.43	1.000
5	1.82	4.85	62.00	96.9728	60.3462	211.487	1.000
6	0.10	4.94	46.21	96.9638	60.0001	270.35	0.944
7	0.10	5.00	44.24	96.9903	60.8904	271.499	0.943
8	0.52	4.86	54.13	97	60	315.341	0.899
9	0.42	4.99	44.87	97	63.9263	324.582	0.890

Tradeoffs must be made to improve one response while accepting a lower standard in another response. The Kalman filter used in this research is a fixed parameter KF. Using response surface methodology, these parameters were optimized. In selecting the KF parameters, the order of importance of the responses is minimizing the number of dead tracks, maximizing the percent correct correlation and maximizing mean prediction percentile. The best parameters that meet this goal are:

Q = 1.98

KG = 4.56

Error Sigma = 62.6

5 Bayesian Inference

Tracking is a statistical estimation problem considered to be best viewed as a Bayesian inference problem. The Bayesian point of view considers additional information such as known information and common sense. Such known information can consist of the aircraft flying dynamics, terrain-target interaction information or a known target goal or mission. Common sense such as a tank cannot traverse a steep incline, or boats avoid land help us to determine additional information that we may wish to incorporate into the model in order to narrow the tracking scope. When this information is processed properly, the possible states of the unknown are more defined. "Bayesian statisticians use prior distribution for unknown parameters even if subjectivity is involved" [Stone, et al., 1999, p.31]. Kalman filtering is cited as a special case of Bayesian filtering [Stone et al., 1999].

The Bayesian inference approach to detection and tracking is not common in the data fusion and tracking community because it is computationally intensive and requires large storage space. Technological advances in computing however allows the exploration of more complex algorithms such as those algorithms using Bayesian inference. One advantage of Bayesian inference is the use of negative information. Kalman filters cannot account for negative information. Negative information requires the existence of a model for the probability of detection. Bayesian inference incorporates prior distribution information as well as the number of targets, their states and how they

change their states. The sensor information is converted into likelihood functions defined on the target space [Stone et al., 1999].

5.1 Use of Likelihood Functions

"Bayesian inference is mathematically precise and uses likelihood functions to represent target tracking information and likelihood functions are the most natural way of representing information added into a tracker. The likelihood functions replace and generalize the notion of contact used in linear Gaussian trackers" [Stone et al., 1999, P.33]. To combine two simultaneous observations with independent errors, multiply the likelihood functions. The likelihood function L for the random variable X and observation Y = y is defined in Equation 5.1.

Equation 5.1 Likelihood Function

$$L(y/x) = \mathbf{Pr}\{Y=y\backslash X=x\} \text{ for } x \in S$$

Where:

Pr = probability density on the measurement space

$x \in S$ = random variable x over the space S

Bayesian inference uses Baye's rule to compute the posterior distribution on the target space. The Bayesian approach has withstood the test of time and is explored as a viable algorithm for tracking multiple targets in a sparse data environment.

5.2 Advantages of Bayesian Inference

The advantages of using Bayesian inference from Stone, et al., 1999, are summarized below. One can see that the use of Bayesian inference appears to be well suited for this problem.

1. Provides a consistent method of reconciling prior and current information.
2. Follows rules and obtains a Baye's optimal estimate
3. Posterior distribution on X given Y = y (current observation) is the starting point
4. Baye’s theorem is naturally recursive

6 Terrain Based Tracking (TBT) Algorithm

Nougues and Brown have developed TBT for ground based targets using local and remote information and discovered that TBT performed slightly better than KF on sample data. Two constraints to the TBT problem were identified as 1) goal regions must be identified and 2) sensor report frequency must allow enough time for the algorithm to produce results prior to receiving the next observation and be able to integrate the targets behavior into its prediction. The TBT algorithm is based on a discrete state-space and uses the ArcInfo GIS software for terrain representation with a resolution of 20 meters per square pixel.

For ground targets moving on roads, the location densities take the shape of the roads. "Trying to fit a mixture of Gaussians to these densities would require an excessive number of kernels to provide a good approximation…at the core of TBT is a set of propagation equations used to model target motion between sensor reports" [Nougues, p.106].

6.1 Problem Statement

Maneuvering targets usually result in a low data rate environment and also exhibit non-Gaussian properties. The terrain based tracker uses Bayesian inference and is an

appropriate tracker when the target or data environment exhibits these properties. Terrain based tracking has been successfully applied to ground based targets. The issue of air target data and the feasibility of terrain based tracking when given air targets is examined. Once the feasibility of applying TBT to air targets is established, the sparse air target data scenario will be applied to terrain based tracking and compared to the Kalman filtering method.

6.2 TBT Model

The filtering function in TBT, Equation 6.1, uses a Bayesian approach in the discrete state space and accounts for the stationary and moving probability densities.

Equation 6.1 TBT Filtering Function and Baye's Rule

$$p^{k+} = p\left[x(k)/z^k\right] = \frac{p[x(k)]p\left[z^k / x(k)\right]}{\int p[x(k)]p\left[z^k / x(k)\right]}$$

Where:

$p\left[z^k / x(k)\right]$ = reported Gaussian density of z^k sensor report given x(k) observation

$p\left[x(k)/ z^k\right]$ = updated density of x(k) observation given z^k sensor report

6.2.1 Benefits of TBT

TBT has benefits that makes the use of terrain based tracking well suited for this problem.

1. The posterior distribution on X given Y = y (current observation) is the starting point
2. Baye's Theorem is naturally recursive

6.2.2 TBT Assumptions

The following assumptions were made in the application of TBT to the air target data set:

1. The relative stationary inertia for the air targets is zero
2. Speed parameter = .46 (This was the calculated average speed parameter)

6.2.3 Probability Distributions

TBT involves non-Gaussian densities. This is ideal for non-linear type tracking and low data rate environments. TBT evaluates the probability of a target moving in one of five directions. Table 6.1 summarizes the five directions.

Table 6.1 Relative Motion Inertia by Direction Difference

Direction	Degrees
P0	0
P1	45
P2	90
P3	135
P4	180

The probabilities differ for each direction based on the underlying terrain. The probabilities for a ground target moving over rough terrain, open terrain, primary or secondary roads, were determined by Nougues for ground target tracking. For this thesis, the probabilities over open and no go, i.e. rough terrain were examined and developed for air targets. These probabilities will be presented later.

7 TBT Modifications for Air Targets

TBT was initially developed for the use of tracking ground targets. The following areas discuss what modifications had to be made in order to perform air target tracking using TBT.

7.1 Combat Information Processor (CIP)

The terrain database was enlarged to incorporate as much of the air target data as possible. The air target data covers much more terrain than the ground targets. The road network and terrain data file was also enlarged on the CIP display as well as within the CIP Tracker program file. The minimum number of cells to expand the grid when tracking air targets also had to be enlarged due to the higher speeds of the air targets and the greater distances being covered between sensor reports.

Additionally, the time projection algorithm was modified by increasing the target speed and decreasing the TBT propagation time step.

7.2 Data Preparation

The air target data covers more terrain than the ground target data over the same time period. The air targets travel inside and out of the terrain box whereas the ground

targets maneuver within the box for the entire time period. Because of the limited size of the terrain box, the air target data traveling outside of this box had to be discarded because TBT does not function outside the terrain box.

The data file was modified as follows. The date-time group was converted into seconds, the z-direction position was eliminated as we are only considering the two dimensional problem, and the velocity in the x and y direction was calculated by Equation 7.1 and Equation 7.2.

Equation 7.1 X-Direction Velocity

$$V_x = \left(\frac{x_2 - x_1}{t_2 - t_1} \right)$$

Equation 7.2 Y-Direction Velocity

$$V_y = \left(\frac{y_2 - y_1}{t_2 - t_1} \right)$$

The data was plotted on the CIP Tracker by plotting the true observations over the entire time period. The ground target reference point was used initially however this caused all of the air targets to appear outside the terrain box. The reference point was therefore adjusted so the air targets maneuvered within the National Training Center road

networks and associated terrain box. The new reference point for the air targets was found to be:

Easting X 490000

Northing Y 3810180

This reference point was validated by plotting five different air target data sets from different days and examining the target paths. This reference point fit all the data sets and seems to be accurate as far as what type of missions the air targets could possibly be carrying out.

A second set of tests was conducted on three sparse data sets. The sparse data sets were created by using data points every 10, 20 and 40 seconds. Sparse data is therefore defined for this experiment as data received greater than or equal to every 10 seconds. A plot of the 10 second data is shown in Figure 7.1:

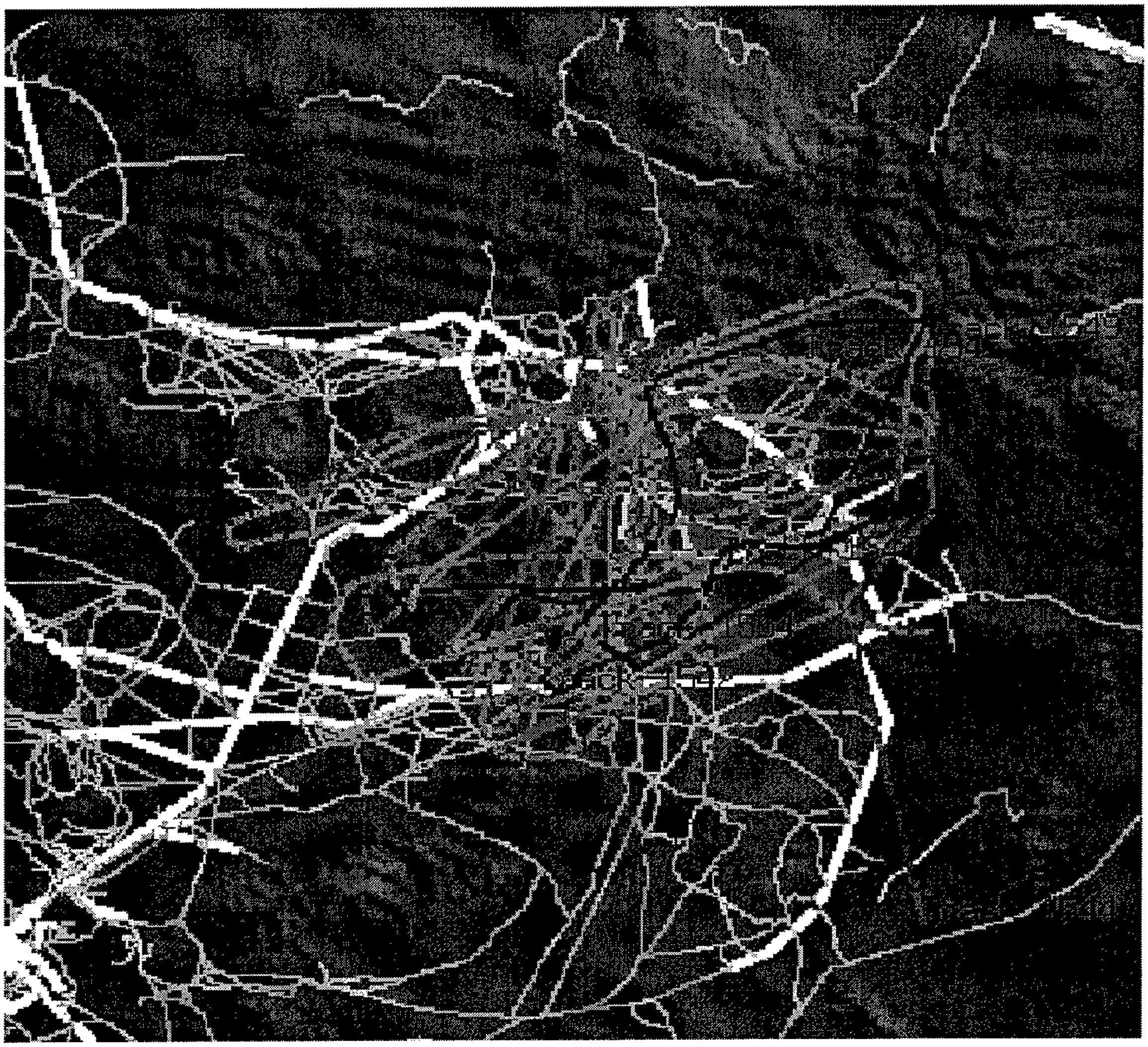


Figure 7.1 Plot of Sparse Data Set

7.3 Parameter Estimation

Two areas of parameter estimation were examined for optimal performance of the terrain based tracker. The first area examined was the parameters for the relative motion inertia by direction difference probabilities. The stationary inertia probability parameters were set to zero for the air targets and the no go and open terrain was considered. Once a

good set of parameters were determined, the error sigma parameter was optimized. Appendix B shows the parameter file used in terrain based tracking.

7.3.1 TBT Relative Motion Inertia by Direction Difference Probabilities

The relative motion inertia by direction difference probability parameters were adjusted for the air targets by running several experiments to determine the set of parameters that yield the best result for tracking the air targets over terrain.

It was found that the air targets, all fixed wing aircraft, do not utilize the road network as frequently as the ground targets, but rather use much of the no go terrain (mountainous terrain) in most cases and try to avoid the open terrain areas by following alongside the no go areas. The targets appeared to be flying into the open areas that are known to be ranges, and are therefore flying into selected open areas when dropping bombs.

The set of parameters found to work best in the air data case are summarized in Table 7.1. These values were developed through extensive experimentation.

Table 7.1 Parameters for Relative Motion Inertia by Direction Difference

P0	P1	P2	P3	P4	Src	Dst
50	40	30	20	10	NO GO	NO GO
50	40	30	20	10	NO GO	OPEN
50	40	30	20	10	NO GO	Sec Road
50	40	30	20	10	NO GO	Pri Road
50	40	30	20	10	Open	NO GO
500	50	10	0	0	Open	Open
400	250	100	20	1	Open	Sec Road
600	450	250	50	1	Open	Pri Road
50	40	30	20	10	Sec Road	NO GO
50	35	20	10	1	Sec Road	Open
1000	500	150	50	1	Sec Road	Sec Road
2000	1000	700	300	5	Sec Road	Pri Road
50	40	30	20	10	Pri Road	NO GO
10	9	8	1	0	Pri Road	Open
150	80	40	10	1	Pri Road	Sec Road
1500	450	150	35	1	Pri Road	Pri Road

7.3.2 TBT Error Sigma Parameter Estimation

Several experiments were conducted to determine the optimal error sigma that causes the best performance for the terrain based tracker. Three replicates at three levels were conducted and the performance using the total number of dead tracks was examined. It was found that the best error sigma for TBT is 80, which resulted in 66 final number of dead tracks for the high data rate environment.

8 Target Tracking

8.1 Single Target Tracking

In order to conduct multiple target tracking, we examine the single target tracking case and break the multiple target problem into single target problems. "The simplest multiple target tracking systems use sequential data processing and nearest neighbor association rule" [Blackman, 1986, P.83]. The basic recursion for single target tracking, Equation 8.1, consists of an initial distribution, the motion update and an information update [Stone, et al., 1999, p. 61].

Equation 8.1 Basic Recursion for Single Target Tracking

(Initial Distribution) $p(t_0, s_0) = q_0(s_0)$ for $s_0 \in S$

For $k \geq 1$ and $s_k \in S$, (Motion Update) $p^-(t_k, s_k) = \int q_k(s_k \setminus s_{k-1}) p(t_{k-1}, s_{k-1}) ds_{k-1}$

(Information Update) $p(t_k, s_k) = \dfrac{1}{C} L_k(y_k \setminus s_k) p^-(t_k, s_k)$

Where:

t_k = Time k

s_k = State k

L_k = Likelihood function

C = Constant that normalizes $p(t_k,\bullet)$ to a probability distribution

y_k = observation k

If no observations are detected than there is no information update. Only a motion update is conducted. The motion update accounts for the transition from t_{k-1} to t_k. The errors or probability distributions do not need to be Gaussian. When they are not Gaussian, we have non-linear tracking.

8.2 Multiple Target Tracking:

Multiple target tracking (MTT) is difficult because we must determine which target generated each radar response. This process is known as data association. Single targets using non-linear tracking methods are applied to the multiple target problem. Data association and state estimation are the principle activities of multiple target tracking [Stone, et al. 1999].

Step 1) Associate contact or observation with current targets.

Step 2) Observations associated with each target are used to estimate the targets' state independent of the other targets.

The extended Kalman filter was formulated to track multiple targets in a high data rate environment. The multiple target tracking problem can be modeled in two parts: 1) the data association problem and 2) The state estimation problem. The number of targets

is estimated and false alarms are determined (False alarms are unlikely when air traffic control is used due to the use of aircraft transponders). The Joint Directors of Laboratories Fusion Sub Panel refers to the process of conducting MTT as level 1 data fusion problem [Stone, et al. 1999].

A non-linear modification to the Kalman filter algorithm was formulated by Yeddanapudi in 1997. This algorithm is called the interactive multiple model (IMM) Kalman filter. Two models considered for its use are the constant velocity model and the maneuver model. This model is used in high data rate environments with small numbers of maneuvers per contact [Stone, et al. 1999].

Typical MTT system assumptions are summarized in Table 8.1 [Stone, et al. 1999, P.26]:

Table 8.1 Classification of Tracking Systems

Target Assumptions		Information Assumptions	
Number	0 - N	Sensor Observations	Linear
		Measurment Error	Gaussian
Motion Model	Gaussian	Type of Sensor data	Contact Output
		False Alarms	yes

9 Sparse Data, Multiple Target Tracking

The same data set analyzed above was transformed into a sparse data set with observations every 10 seconds. Both KF and TBT were run and the best parameters were found that yielded the best results in the response variable mean prediction percentile, percent correctly correlated and number of dead tracks.

9.1 Kalman Filtering of Sparse Data

The same procedure to optimize the KF parameters was repeated using response surface methodology. The same tests were run with the same design of experiment from the tests performed with the full data set. The optimization was then performed on the three responses.

Figure 9.1 shows the Kalman filter performing sparse data tracking over the 10 second interval data. Figure 9.1 shows four tracks. The total length of the path indicates the correlated movement of the target. The rectangles are the Kalman filter prediction gates.

Figure 9.1 Plot of Kalman Filtering and Sparse Data Tracking

9.1.1 Response = Percent Correctly Correlated

The reduced model obtained from stepwise regression was examined. It was found that the interaction between Q and error sigma was significant. As a result, both Q and error sigma were added back into the model in order to maintain a hierarchical model. It was found that Q and error sigma alone are not significant in the model, however their interaction is significant as found as a result of conducting a t-test. This model is as follows:

Factor	Coefficient Estimate	DF	Prob > |t|
Q	0.72	1	0.6882
KG	2.66	1	0.1493
Error Sigma	4.75	1	0.0135
Q * KG	4.57	1	0.0379

The best model for percent correctly correlated is summarized in Table 9.1. The adjusted $R^2 = 0.277$, however our model is significant as the Prob > F is less than 0.05.

Table 9.1 ANOVA for Percent Correctly Correlated Sparse Data

Source	Sum of Squares	DF	Mean Square	F Value	Prob > F
Model	815.43	4	203.86	3.48	0.0238
Residual	1287.10	22	58.50		
Cor Total	2102.53	26			
Root MSE	7.65		R-Squared	0.3878	
Dep Mean	54.56		Adj R-Squared	0.2765	
C.V.	14.02		Pred R-Squared	0.0693	
PRESS	1956.74		Adeq Precision	7.280	Desire > 4

The parameter Q contributes most to our model. The final equation, Equation 9.1, in terms of the original factors is:

Equation 9.1 Percent Correct Corelation in Terms of Actual Factors:

$$\% \text{ Correctly Correlated} = 32.22 + 9.68\ Q + 4.4\ KG + 0.16\ \text{Error Sigma} - 2.75\ Q * KG$$

Figure 9.2 shows the relationship between Q, KG, error sigma and percent correct correlation. In order to have the highest percent correctly correlated we should select Q and error sigma at the high level and KG at the low level or KG and error sigma high and Q low. Table 9.2 summarizes the optimal solutions obtained for the respective values of Q, KG and error sigma in order to have the highest percent correctly correlated.

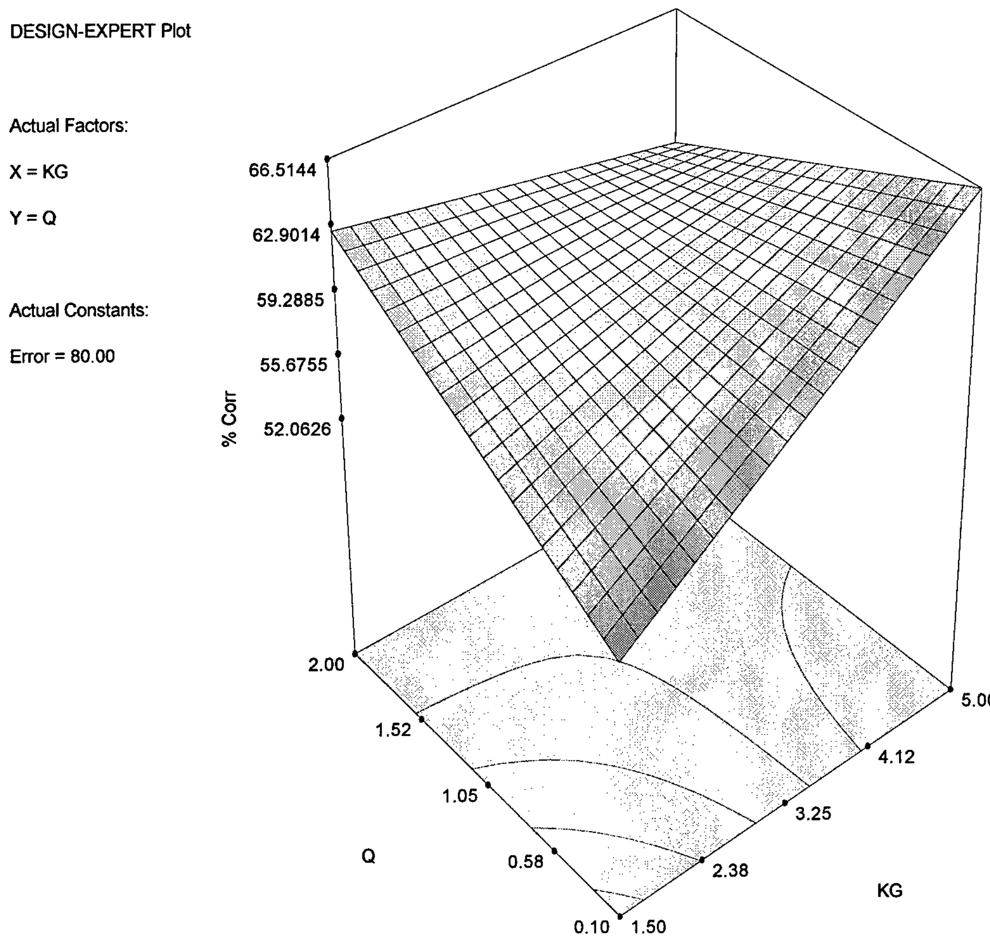


Figure 9.2 Percent Correct Correlation Response Surface

Table 9.2 Sparse Data KF Optimal Solutions

	Q	KG	Error Sigma	% Corr	Desirability
1	0.10	5.00	80.00	66.5143	0.941
2	0.11	5.00	80.00	66.4687	0.940
3	0.10	5.00	79.13	66.3765	0.938
4	0.10	4.91	80.00	66.1482	0.932
5	0.25	4.69	80.00	64.7686	0.896
6	0.18	4.42	80.00	63.9337	0.874
7	0.62	4.74	80.00	63.7034	0.869
8	2.00	1.50	79.92	62.6147	0.840

9.1.2 Response = Mean Prediction Percentile

The reduced model, obtained from using stepwise regression, was examined.

Factor	Coefficient Estimate	DF	Prob > |t|
Q	2.79	1	0.0097
Error Sigma	1.09	1	0.2974
Q * Error Sigma	3.22	1	0.0120

The KG parameter was found not to be significant in the model predicting mean prediction percentile. Table 9.3 verifies that the model selected is a good fit as Prob > F = 0.006. The model has an adjusted $R^2 = 0.335$.

Table 9.3 ANOVA for Mean Prediction Percentile Sparse Data

Source	Sum of Squares	Mean DFSquare	F Value	Prob > F
Model	301.10	3100.37	5.37	0.0060
Residual	429.67	2318.68		
Cor Total	730.78	26		

Root MSE	4.32	R-Squared	0.4120	
Dep Mean	20.79	Adj R-Squared	0.3353	
C.V.	20.79	Pred R-Squared	-0.1308	
PRESS	826.38	Adeq Precision	7.233	Desire > 4

Equation 9.2 shows the model in terms of the actual factors. The Q parameter contributes most to the prediction of mean prediction percentile.

Equation 9.2 Mean Prediction Percentile in Terms of Actual Factors:

Mean Prediction Percentile = 21.56 – 2.71 Q –0.083 Error Sigma + 0.11 Q * Error Sigma

Figure 9.3 shows the relationship between Q and the number of dead tracks. It is noted that KG has no affect on the model. In order to achieve the maximum mean percent prediction, Q and error sigma should be selected at the high level.

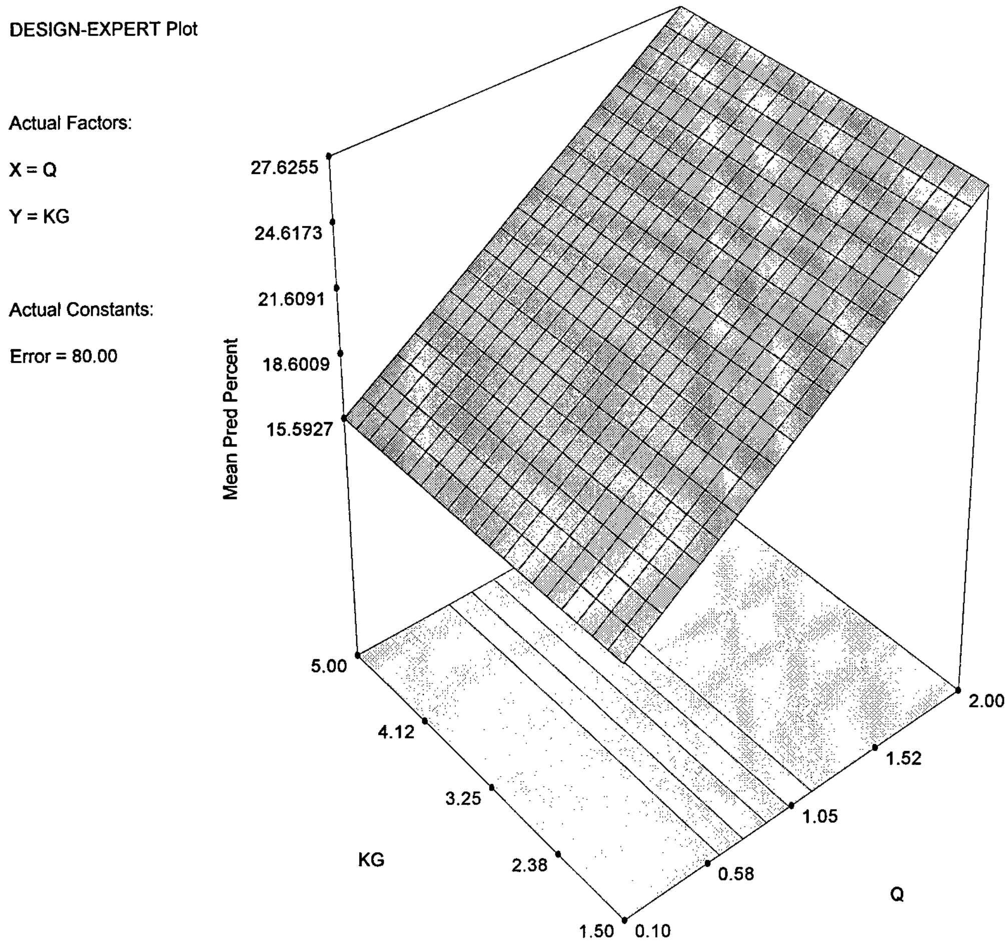


Figure 9.3 Mean Prediction Percentile Response Surface

Table 9.4 shows the optimal solutions for the given values of Q, KG and error sigma.

Table 9.4 Mean Prediction Percentile Optimal Solution

	Q	KG*	Error Sigma	Mean Pred Percent	Desirability
1	1.97	*1.81*	78.51	27.2236	1.000
2	1.98	*3.85*	78.46	27.2929	1.000
3	1.97	*1.53*	79.73	27.4199	1.000
4	1.99	*4.32*	78.02	27.2852	1.000
5	1.97	*4.71*	79.65	27.3761	1.000
6	1.95	*2.36*	77.97	27.0087	1.000
7	1.91	*4.03*	79.86	27.0619	1.000
8	1.97	*3.19*	78.01	27.1759	1.000
9	0.10	*1.57*	20.00	19.8669	0.743
10	0.10	*2.64*	20.00	19.8669	0.743

*Has no effect on optimization results.

9.1.3 Response = Number of Dead Tracks

The reduced model, obtained from using stepwise regression, was examined.

Factor	Coefficient Estimate	DF	Prob > |t|
Error Sigma	-40.89	1	0.0021

It was found that error sigma is the only significant factor in predicting the number of dead tracks. The best model is summarized in Table 9.5 and has an adjusted $R^2 = 0.293$.

Table 9.5 ANOVA for Number of Dead Tracks Sparse Data

Source	Sum of Squares	DF	Mean Square	F Value	Prob > F
Model	31214.88	1	31214.88	11.79	0.0021
Residual	66173.79	25	2646.95		
Cor Total	97388.67	26			

Root MSE	51.45	R-Squared	0.3205	
Dep Mean	257.78	Adj R-Squared	0.2933	
C.V.	19.96	Pred R-Squared	0.1884	
PRESS	79041.76	Adeq Precision	5.841	Desire > 4

Equation 9.3 shows the model in terms of the actual factors. Error sigma is the only factor used to predict the number of dead tracks.

Equation 9.3 Number of Dead Tracks in Terms of Actual Factors:

 # Dead Tracks = 321.39 – 1.36 Error Sigma

Figure 9.4 shows the relationship between error sigma and the number of dead tracks.

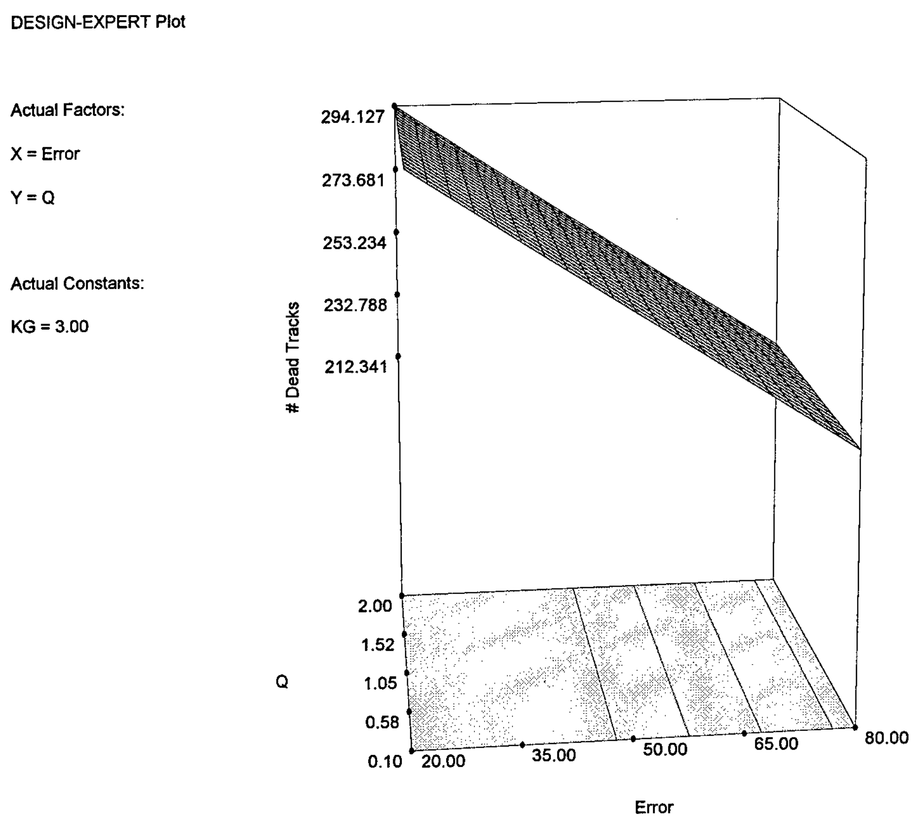


Figure 9.4 Number of Dead Tracks Response Surface

Table 9.6 shows the optimal solutions for the given values of error sigma:

Table 9.6 Optimal Solutions

	Q*	KG*	Error Sigma	# Dead Tracks	Desirability
1	*1.68*	*1.56*	80.00	212.341	0.874
2	*0.47*	*4.62*	80.00	212.341	0.874
3	*0.80*	*2.64*	80.00	212.341	0.874
4	*1.84*	*2.62*	80.00	212.341	0.874
5	*1.55*	*2.36*	80.00	212.341	0.874
6	*1.36*	*3.12*	80.00	212.341	0.874
7	*1.09*	*2.18*	80.00	212.341	0.874
8	*1.86*	*1.56*	80.00	212.341	0.874
9	*1.44*	*2.53*	80.00	212.341	0.874
10	*1.34*	*3.88*	80.00	212.341	0.874

*Has no effect on optimization results.

Optimizing for all three response variables yields the solutions found in Table 9.7. The best two solutions, solution 1 and 5, show error sigma at the high level, Q at the high and low level and KG at the low and high level.

Table 9.7 KF Parameters Optimal Solutions

	Q	KG	Error Sigma	% Corr	Mean Pred	# Dead Tracks	Desirability
1	2.00	1.50	80.00	62.628	27.6254	199.973	0.884
2	2.00	1.50	79.48	62.5403	27.5448	200.708	0.882
3	1.18	5.00	80.00	62.1466	22.42	200.79	0.828
4	1.86	4.65	80.00	59.6248	26.7502	217.799	0.826
5	0.74	5.00	80.00	63.9364	19.6228	188.778	0.825

When in a sparse data environment, KF doesn't do a very good job as expected, however the best results can be achieved using the following parameters:

Q = 0.74

KG = 5

Error Sigma = 80

This is a slight change from the high data rate parameters where we found Q to be at the high level.

9.2 Terrain Based Tracking of Sparse Data

The full data set was made sparse by keeping the observations at 10, 20 and 40 second intervals. Figure 9.5, Figure 9.6, and Figure 9.7 show terrain based tracking at the 10 second interval.

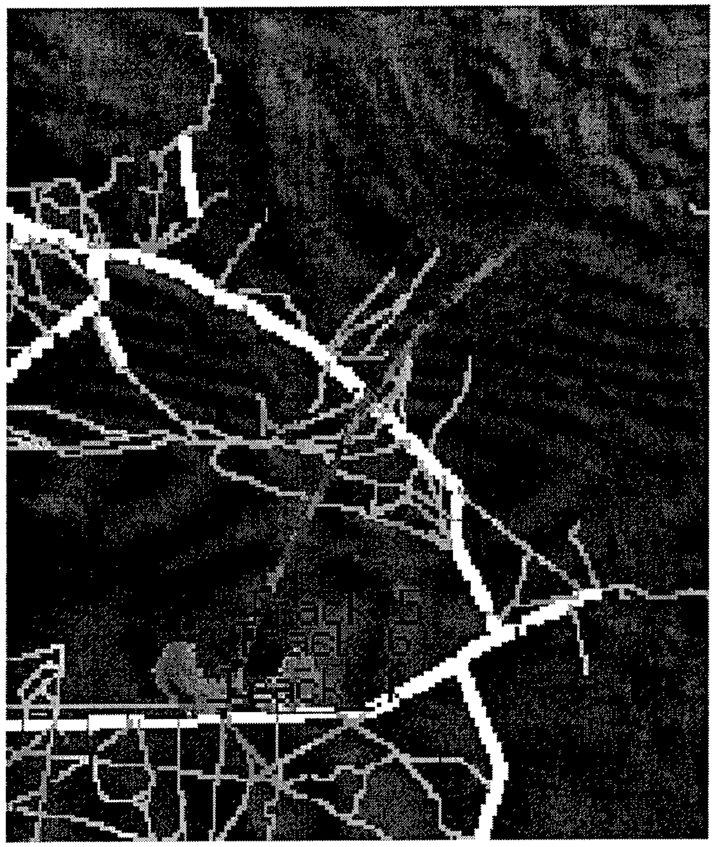


Figure 9.5 TBT and Sparse Data Tracking #1

These figures show two targets traveling together. Figure 9.5 shows that we are about to lose correlation with track 4 as we see the initiation of track 6.

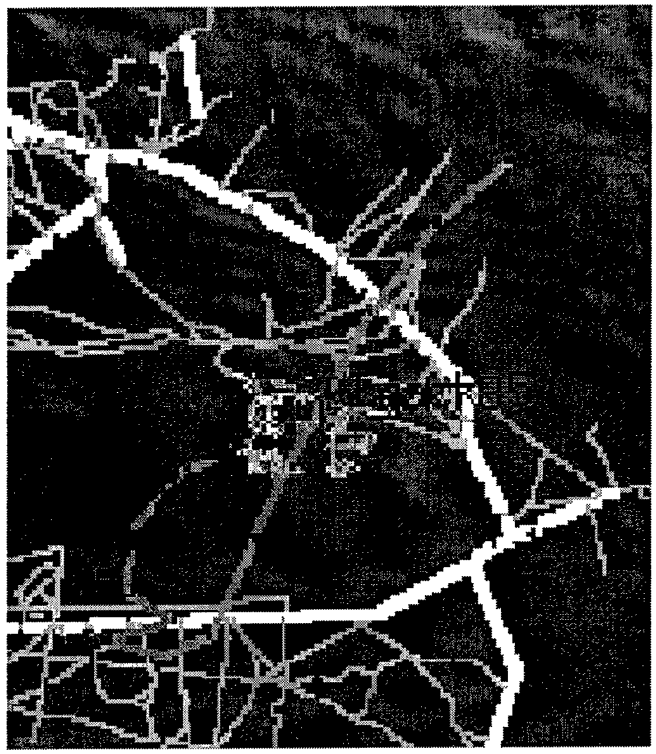


Figure 9.6 TBT and Sparse Data Tracking #2

Figure 9.6 shows that we are still tracking track 5 and we are about to lose correlation with track 6. A new track is being initiated as track 7. Figure 9.7 shows that we have tracked track 5 without having to initiate any new tracks and we have tracked the other target with only initiating one additional track

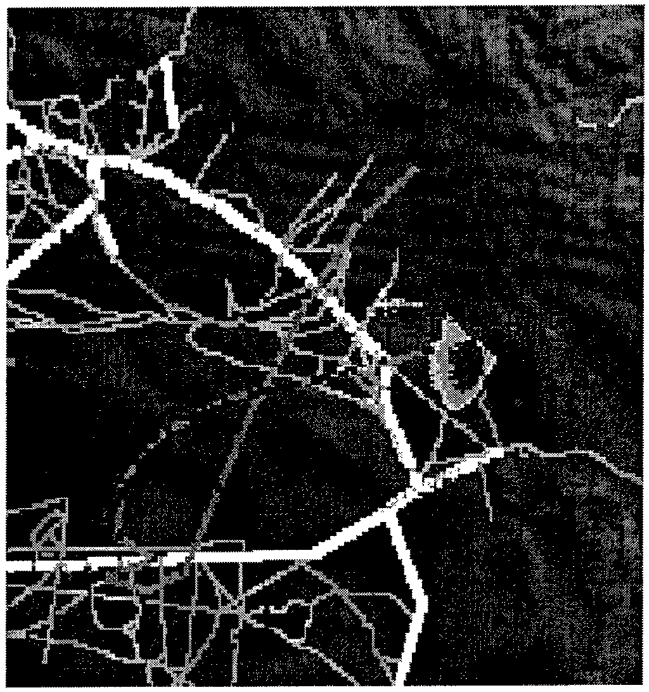


Figure 9.7 TBT and Sparse Data Tracking #3

It is important to utilize the percent correctly correlated statistic and mean prediction percentile when comparing results across the different data sets and using one methodology. The sparse data set has a reduced number of sensor reports and therefore will always result in fewer numbers of dead tracks. However, when comparing the same data set between the two methodologies, we make use of all three statistics. TBT has the ability to maintain track correlation over an extended time period while receiving sparse sensor reports. We could also compute the percentage of dead tracks and use this statistic for our comparison.

10 Performance Evaluation of the Tracking Systems

Table 10.1 summarizes the performance of the Kalman filter and the terrain based tracker for the full data set. It can be seen that the TBT method has a slightly higher percent correct correlation and much lower number of dead tracks, and a lower mean prediction percentile. Ideally, the percent correct correlation and mean prediction percentile should be as high as possible and the number of dead tracks should be as low as possible.

Table 10.1 Tracker Performance Summary with Full Data Set

Tracker	%Correct Correlation	Mean Prediction Percentile	# of Dead Tracks
Kalman filter	97.0%	90.0	144
TBT	98.6 %	31.6	66

10.1 Kalman Filtering Correlation

The Kalman filter performed below that of the terrain based tracker when using the complete data set and the sparse data set. This was as expected given the non-linear,

non-Gaussian nature of the data set and the known fact that Kalman filtering is not well suited for this type of data. KF correlations were well over 95% when using the full data set. The KF methodology produced 144 dead tracks. This is poor performance considering the data set had only 10 real targets in the data set.

10.2 Terrain-Based Tracking Correlation

The terrain based tracker performed slightly better in percent correctly correlated and significantly better in the number of dead tracks. TBT always performs lower in mean prediction percentile than KF. This is due to the fact that TBT considers movement in one of five directions and utilizes intelligent target behavior which leads to a higher percent correctly correlated. The mean prediction percentile for TBT is lower than that for Kalman filtering in all cases [Bovey, 2000].

10.3 Conclusions

The terrain based tracker out performed the Kalman filter in predicting target location in a high data rate environment. While this is a strength of Kalman filtering, the benefits of applying Bayesian inference and utilizing the underlying terrain provides a greater overall benefit to target tracking than that which Kalman filtering can provide.

11 Sparse Data Performance Evaluation of the Tracking Systems

Table 11.1 summarizes the performance of the Kalman filter and the terrain based tracker for the sparse data set where sensor reports were received every 10 seconds. Additionally, four tracks were removed from the data set because they were located on the edge of our terrain file. Given the sparse data scenario, TBT continues to propagate until the next observation is received. The targets near the edge of the terrain file were propagating outside the terrain file causing TBT to shut down. To avoid this problem, we eliminated the targets at the edge of the terrain data file.

Table 11.1 Tracker Performance Summary with 10 Sec Sparse Data Set

Tracker	%Correct Correlation	Mean Prediction Percentile	# of Dead Tracks
Kalman filter	63.93%	19.62	188
TBT	66.85%	1.35	151

It was found that TBT out performed Kalman filtering once again. TBT has a slightly higher percent correct correlation and 29 fewer dead tracks.

Next the data file was made sparse, every 20 seconds. Table 11.2 summarizes the performance of the Kalman filter and the terrain based tracker for the 20 second sparse data set.

Table 11.2 Tracker Performance Summary with 20 Sec Sparse Data Set

Tracker	%Correct Correlation	Mean Prediction Percentile	# of Dead Tracks
Kalman filter	52.99%	31.35	105
TBT	63.59%	2.57	83

A slight drop in the percent correctly correlated is seen with TBT when going from 10 seconds to 20 seconds. Kalman filtering however has a more significant change in the percent correctly correlated when comparing 10 and 20 second interval data sets.

Next the data file was made sparse, every 40 seconds. Table 11.3 summarizes the performance of the Kalman filter and the terrain based tracker for the 40 second sparse data set.

Table 11.3 Tracker Performance Summary with 40 Sec Sparse Data Set

Tracker	%Correct Correlation	Mean Prediction Percentile	# of Dead Tracks
Kalman filter	34.59%	0%	73
TBT	58.92%	9.66	46

Next the data file was made sparse, every 60 seconds. Due to the small number of data points remaining, 115, no correlations could be made using KF or TBT.

11.1 Kalman Filtering Correlation

The performance of the KF falls as the interval between observations increases from 10 to 20 to 40 seconds. This verifies what was learned from reading the literature and shows that Kalman filtering is not well suited for the sparse data environment.

11.2 Terrain Based Tracking Correlation

The TBT parameter file was adjusted for the sparse data set in order to perform tracking in real time. The full data set performed TBT in 0.1 second time step intervals. This worked fine given the high data rate of one second. When utilizing the sparse data

sets however, the time step had to be increased in order to keep the prediction process ahead of real time. It was found that 0.1 second intervals caused the tracker to slow down considerably in order to propagate the densities every 0.1 time steps. This parameter was changed to 0.8 second time steps and resulted in a decreased prediction time so that TBT predicts ahead of the next observation being received. This change in parameter did not significantly affect the performance of TBT.

It can be seen that the performance of TBT falls slightly from the 10 second observation interval to the 20 second interval, however TBT still out performs the KF method. This verifies that a Bayesian inference method is a viable method and improves upon the current tracking methods.

11.3 Conclusions

Terrain based tracking helps improve tracking ability given both a high and low data rate environment over the Kalman filter methodology when using the sample data sets. Table 11.4 and Table 11.5 summarizes the performance of TBT and KF given the three sparse data files. No results were obtained for the 40 second interval, error = 80 in TBT. This is most likely due to propagating outside the terrain box.

Table 11.4 TBT Sparse Data Performance

TBT	10 Second Interval			20 Second Interval			40 Second Interval		
Error Sigma	% Corr correlation	Mean Prediction Percentile	# Dead Tracks	% Corr correlation	Mean Prediction Percentile	# Dead Tracks	% Corr correlation	Mean Prediction Percentile	# Dead Tracks
20	60.06	0.72	184	57.07	1.67	100	57.3	9.16	52
40	61.72	0.96	172	58.42	2.74	96	58.92	9.66	46
80	66.85	1.35	151	63.59	2.57	83	No Results		

Table 11.5 KF Sparse Data Performance

KF	10 Second Interval			20 Second Interval			40 Second Interval		
Error Sigma	% Corr correlation	Mean Prediction Percentile	# Dead Tracks	% Corr correlation	Mean Prediction Percentile	# Dead Tracks	% Corr correlation	Mean Prediction Percentile	# Dead Tracks
20	55.76	20.00	252	39.95	0.56	169	30.27	0	85
40	63.11	20.14	205	45.65	21.28	128	31.89	0	82
80	68.79	26.75	162	54.08	22.42	101	34.59	0	73

Figure 11.1 and Figure 11.2 depicts the graphical results

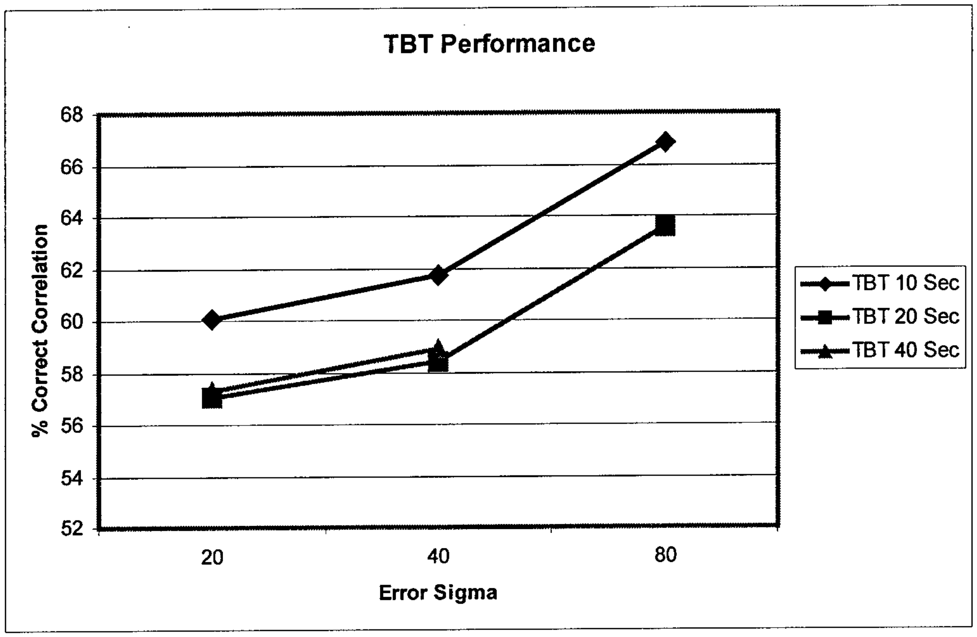


Figure 11.1 TBT Graphical Performance

Better results are obtained in the percent correct correlation in TBT and KF as the error sigma is increased.

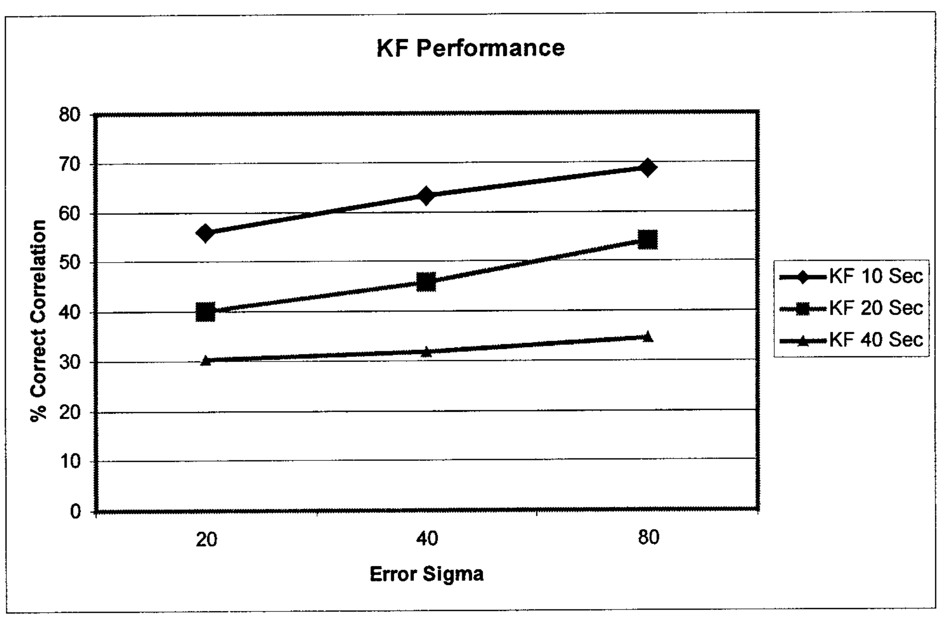


Figure 11.2 KF Graphical Performance

In comparing TBT and KF, it can be seen that as we increase the error sigma the number of dead tracks decreases for both methods. Figure 11.3 shows us how TBT conducts sparse data tracking with fewer number of dead tracks than KF as we change error sigma.

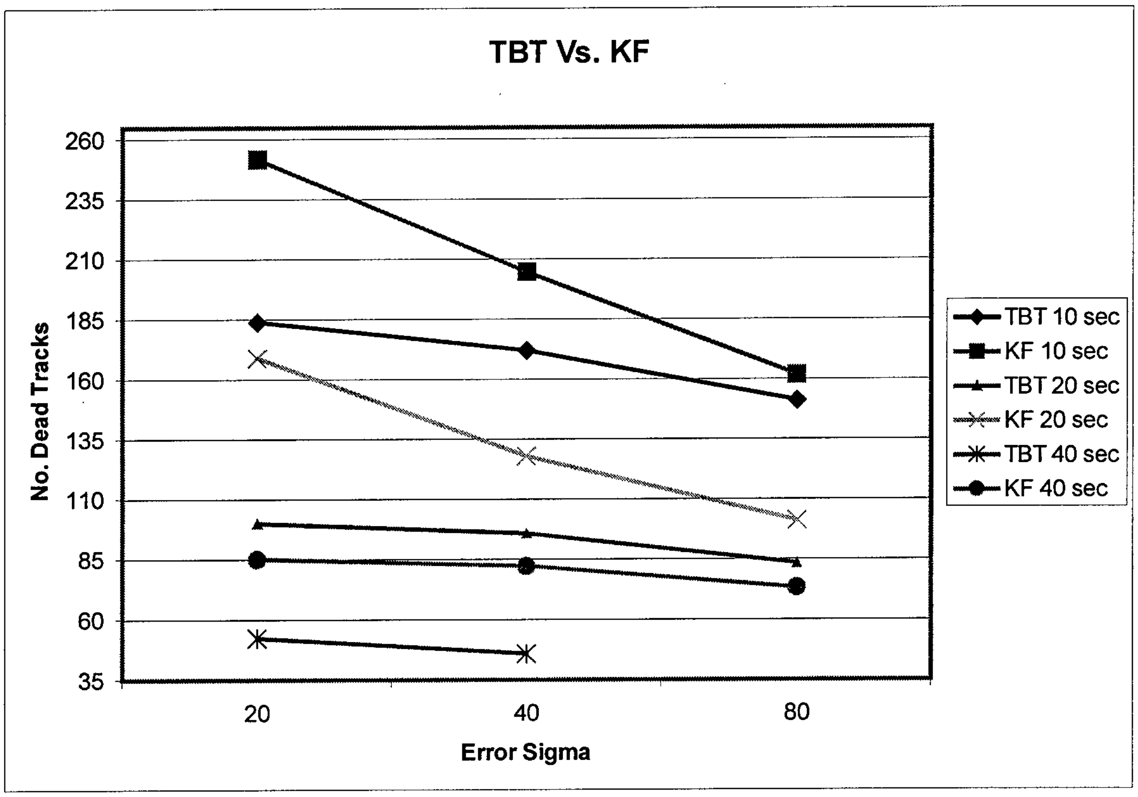


Figure 11.3 TBT Vs KF Performance

12 Summary and Conclusions

12.1 Summary

The TBT algorithm is a viable method of tracking air targets in the future given the computing power of today's computers. More importantly, these computers will fit in existing hardware. While terrain based tracking and Bayesian inference is currently used in the Navy and Coast Guard systems, there is also future potential use for terrain based tracking and Bayesian inference in air target tracking for military and civilian use.

12.2 Contributions

This thesis contributes to the knowledge base of multiple target tracking in the following areas:

1. The development of a test environment
2. The extension of the terrain based tracker for the use of tracking air targets
3. The optimization of Kalman filtering parameters using response surface methodology
4. The test and evaluation of Kalman filtering and terrain based tracking

Two test environments were developed; a high data rate environment and a sparse data environment. These environments were simulated on an Ultra-2 Unix computer given a terrain database represented in the ArcInfo GIS software, the combat information processor and the cipTracker program.

The TBT program was modified in order to support the tracking of air targets. First, the goal regions for air targets were identified. A new reference point was found to apply to the air target data. Probabilities for an air target operating in no go and open terrain were developed and tested over multiple air target data sets and the stationary inertia probabilities were set to zero. Next, the CIP was modified to incorporate the entire terrain database and the expanded road network. Finally, it was found that TBT is optimized for high levels of error sigma.

Response surface methodology was used to optimize the KF parameters in both the high data rate environment and the sparse data environment. All three responses were modeled in both environments. The high data rate environment resulted in models with higher adjusted R^2. The sparse data environment models resulted in lower adjusted R^2 but we could still find relationships between the factors and were able to optimize the parameters. The optimal parameters for each environment differed slightly in parameter Q. The high data rate environment found Q at the high level to be optimal, while the sparse data environment found Q at the low level to be optimal. This is not significant as we are comparing KF to TBT and are really interested in the number of dead tracks when

comparing the two methods. Error sigma was the only significant KF parameter when minimizing the number of dead tracks in the sparse data environment model.

The testing and evaluation of KF and TBT was conclusive in finding that TBT performs better than KF in both the high data rate environment and the sparse data environment given the sample data sets. Tests were run over 10, 20, and 40 second data intervals and found that KF performance is reduced significantly as the interval between sensor reports is increased. TBT on the other hand performs consistently in spite of the increasing interval between sensor reports.

12.3 Recommendations

It is recommended that the TBT algorithm be modified to incorporate maneuver adaptive filtering. This adaptive mode would allow for two or more correlation regions thus making the algorithm more flexible. For instance, if the target starts to maneuver, the adaptive filter could take this into account and apply a different set of parameters and probabilities. Much of the literature on target tracking discusses the ability to use adaptive tracking parameters. This is a potential area for improving TBT as well. The table of TBT probabilities may be very good parameters for some targets. Another set of targets may perform better given a different set of probabilities.

This area also lends itself to the topic of optimizing the TBT parameters for direction difference probabilities. There are 14 different categories of traveling from one area to another by 5 different direction parameters, for a total of 70 parameters. These parameters have been determined through experimentation and prior knowledge of target movement over terrain. Perhaps an optimization technique suited for a large number of parameters could be applied to find the optimal set of parameters.

12.4 Areas for Future Research

From a systems engineering perspective, this work contributes a novel application of terrain based tracking as applied to air targets. As three-dimensional representation in computing improves, future research should explore the three dimensional air target data and apply it to terrain based tracking.

Another area for future research is through group correlation analysis of air targets. In many instances, air targets travel in groups. Research by Brandon Bovey in the group correlation of ground targets could be applied to the air target data.

This research should also examine the concept of goal regions. These are areas known to be of significant importance and are areas that a target is highly likely to travel to, in or around. For instance, in the air target tracking problem, potential areas for goal regions may include known air defense radar sites, ammunition depots, supply points,

range facilities, and airfields. Incorporating these areas would be approached the same way road networks are incorporated into TBT. These additional areas would also use Bayesian inference when a target approaches a known or potential goal region.

Finally, TBT could also be used in conjunction with a KF algorithm or some other type of tracking algorithm that could improve tracking performance of the TBT used alone. This area would lend itself to classifying a target as linear and applying a linear tracker and when classified as a nonlinear track, apply TBT or a similar Bayesian inference or non-linear tracker.

References

1. Ackoff, Russell L. *A Manager's Guide to Operations Research*, John Wiley and Sons, Inc., 1963.

2. Bar-Shalom, Yaakov. *Multitarget-Multisensor Tracking: Applications and advances*, Volume II., Norwood, Massachusetts: Artech House, Inc., 1992.

3. Blackman, Samuel S. *Multiple-Target Tracking with Radar Applications*, Norwood, and Massachusetts: Artech House Inc., 1986.

4. Bovey, Brandon and Brown, D.E. *Target Tracking Prediction Through Group Correlation Analysis,* Department of Systems Engineering, University of Virginia, 2000.

5. Brookner, Eli. *Tracking and Kalman Filtering Made Easy*. John Wiley and Sons, Inc, 1998.

6. Cipra, Barry, (1993).: http://www.siam.org/siamnews/mtc/mtc893.htm.

7. Crouch, Ingrid W. M., Greenwood, Allen G. and Rees, Loren Paul. *Use of a Classifier in a Knowledge-Based Simulation Optimization System*. Naval Research Logistics, Vol. 42, pp 1203-1232, John Wiley and Sons, Inc., 1995.

8. Derringer, George and Suich, Ronald. *Simultaneous Optimization of Several Response Variables*. Journal of Quality Technology, Vol. 12, No. 4, October 1980.

9. Gros, X.E. *NDT Data Fusion*. John Wiley and Sons, Inc.,1997.

10. Montgomery, Douglas C. and Bettencourt, Vernon M., Jr. *Multiple Response Surface Methods in Computer Simulation*. ORSA/TIMS National Meeting, Miami, Florida, November 1976.

11. Myers, Raymond H. and Montgomery, Douglas C. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley and Sons, Inc, 1995.

12. Nougues, Pierre-Olivier, *Tracking Objects in Terrain*, Ph.D. Dissertation, University of Virginia, January, 1996.

13. Nougues, Pierre-Olivier and Brown, Donald E. *We Know Where You Are Going: Tracking Objects in Terrain.* IMA Journal of Mathematics Applied in Business and Industry, 1997.

14. Robinson, Guy. (1995). http://www.npac.syr.edu/copywrite/pcw/node464.htm.

15. Ramoni, Marco and Sebastiani, Paola. *Bayesian Methods for Intelligent Data Analysis*, Knowledge Media Institute, KMI Technical Report, July 1998.

16. Space Analytics Associates, Inc., (1999). http://www.sni.net/saa/short.htm .

17. Stone, Lawrence D., Barlow, Carl A. and Corwin, Thomas L. *Bayesian Multiple Target Tracking*, Artech House,1999.

18. Todling, R. and Cohn, S.E. *Some Strategies for Kalman Filtering and Smoothing.* USRA/NASA Data Assimilation Office, Greenbelt, U.S.A.

19. Vining, G. Geoffrey. *A Compromise Approach To Multi-response Optimization.* Journal of Quality Technology, Vol. 30, No.4, October 1998.

20. Wagner , Daniel H. and Associates. (1999). http://www.wagner.com/technologies/search/ casp2.htm.

21. Washburn, Alan R., *Search and Detection*, ORSA Books, 2nd Edition, November 1982.

22. Welch, Greg and Bishop, Gary. *An Introduction to the Kalman Filter*, Department of Computer Science, University of North Carolina-Chapel Hill, TR 95-041, October 19,1998.

Appendix A: Kalman Filter Sample Parameters Data File

```
# ---------------------------------------------------------------------
#           Configuration file for the CIPTracker                    #
# ---------------------------------------------------------------------
# note: to draw the NTC area launch app with -ntc flag
#
# turn off logging and graphics for fastest runs
LOGGING_OFF
GRAPHICS_OFF
# file with time-ordered data...
SENSORDATA data/15Jan_Test_Set.txt
T 1 # seconds between new observations
ERRORSIGMA      20      #default error sigma = 100
# observation information...
EASTINGX    490000
NORTHINGY 3810180               # sensor report error
# file containing AHAS vehicle info (esit table)
# VEHICLEDATA     data/esit.ascii
# track maintenance and correlation parameters...
CORRTHRESH      .00000000001 # threshold for NN correlation
DELOLDTRACK     20              # period of time (seconds) to keep stale tracks
```

```
# display parameters...

GRIDDRAWTHRESH          .25     # draw probabilities above this proportion of mass

MAXTRACKDISPLAYLENGTH   9999 # the maximum display length (#obs) of tracks

NUMGRIDCOL      13      # max number of grid colors to display from each palette

NUMGRIDCOLFILES         4       # number of grid color palettes available

SCALEFACTOR     10.0            # (wall time) / (sim time)


# parameters for the Kalman filter...

KALMANT    1     # interval between reports

KALMANQ .1

KALMANKG 5

# -----------------------------------------------------------------------
```

Appendix B: Terrain Based Tracking Parameters Data File

```
#
# TBT Parameters
# COMBINED FILE
# data/combined_file_90mCells_662x506.raw
/home/bjb9d/research/cipTracker/data/combined_file_90mCells_1013x842.raw


# NW CORNER UTM COORDS
492800 3955480


# CELL SIDE LENGTH
90 # grid size
# 635 480 # width (columns) and height (rows) of 90 meter cell file
1013 842


.46                 # Speed units (m/s)
.1                  # time step in seconds
10                  # max time steps to cross cell (limits array size)
60                        # min number of cells to expand grid


1            # time evolution factor (sim time/wall time)
0.02   # Psm (prob. stationary to moving)
```

```
0.01     # Pms (moving to stationary)
#
# relative motion inertia by direction difference
#
# p0  p1  p2  p3  p4 #      Src    Dst
  50  40  30  20  10 #      No Go  No Go
  10   9   8   1   0 #    No Go  Open
  50  40  30  20  10 #      No Go  SecRoad
  50  40  30  20  10 #      No Go  PriRoad
  10   9   8   1   0 #     Open   No Go
 500  50  10   0   0 # 22% Open    Open
# 500  80  20   5   1 # 22% Open    Open
 400 250 100   5   0 # 28% Open    SecRoad
 600 450 250  50   1 # 50% Open    PriRoad
  50  40  30  20  10 #      SecRoad No Go
  50  35  20  10   0 # 2% SecRoad Open
1000 500 150  50   1 # 29% SecRoad SecRoad
2000 1000 700 300  5 # 69% SecRoad PriRoad
  10   9   8   1   0 #    PriRoad No Go
  10   9   8   1   0 # 1% PriRoad Open
 150  40  30  20  10 # 11% PriRoad SecRoad
1500 450 150  35   1 # 87% PriRoad PriRoad
```

```
#

# Relative stationary inertia
  # Src    Dst
 0 # No Go   No Go
 0 # No Go   Open
 0 # No Go   SecRoad
 0 # No Go   PriRoad
 0 # Open    No Go
 0 # Open    Open
 0 # Open    SecRoad
 0 # Open    PriRoad
 0 # SecRoad No Go
 0 # SecRoad Open
 0 # SecRoad SecRoad
 0 # SecRoad PriRoad
 0 # PriRoad No Go
 0 # PriRoad Open
 0 # PriRoad SecRoad
 0 # PriRoad PriRoad
END_OF_PARAMETERS
```